JEDEC STANDARD

TS511X, TS521X Serial Bus Thermal Sensor Device Standard

JESD302-1A

Revision 2.00

(Revision of JESD302-1.01, April 2022)

August 2023

JEDEC SOLID STATE TECHNOLOGY ASSOCIATION

NOTICE

JEDEC standards and publications contain material that has been prepared, reviewed, and approved through the JEDEC Board of Directors level and subsequently reviewed and approved by the JEDEC legal counsel.

JEDEC standards and publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for use by those other than JEDEC members, whether the standard is to be used either domestically or internationally.

JEDEC standards and publications are adopted without regard to whether or not their adoption may involve patents or articles, materials, or processes. By such action JEDEC does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the JEDEC standards or publications.

The information included in JEDEC standards and publications represents a sound approach to product specification and application, principally from the solid state device manufacturer viewpoint. Within the JEDEC organization there are procedures whereby a JEDEC standard or publication may be further processed and ultimately become an ANSI standard.

No claims to be in conformance with this standard may be made unless all requirements stated in the standard are met.

Inquiries, comments, and suggestions relative to the content of this JEDEC standard or publication should be addressed to JEDEC at the address below, or refer to www.jedec.org under Standards and Documents for alternative contact information.

Published by
©JEDEC Solid State Technology Association 2023
3103 North 10th Street
Suite 240 South
Arlington, VA 22201-2107

PRICE: Contact JEDEC

Printed in the U.S.A.

DO NOT VIOLATE
THE
LAW!

JEDEC Solid State Technology Association
3103 North 10th Street
Suite 240 South
Arlington, VA 22201-2107
https://www.jedec.org/contact

This page intentionally left blank

TS511X, TS521X Serial Bus Thermal Sensor Device Standard

Contents

Contents (cont'd)

Contents (cont'd)

List of Figures

Contents (cont'd)

List of Tables

Contents (cont'd)

Contents (cont'd)

DEFINITION of TS521X, TS511X THERMAL SENSING DEVICE for MEMORY MODULE APPLICATIONS

(From JEDEC Board Ballot JCB-23-25, formulated under the cognizance of the JC-40.1 Subcommittee on Digital Logic Families and Applications, committee item # 401.04).

1 Scope

This standard defines the specifications of interface parameters, signaling protocols, and features for fifth generation Temperature Sensor (TS5) as used for memory module applications. These device operate on I^2C and I3C two-wire serial bus interface. The designation TS521X and TS511X refers to the device specified by this document.

The purpose is to provide a standard for the TS5 device for uniformity, multiplicity of sources, elimination of confusion, ease of device specification, and ease of use.

Unless otherwise noted in the document, any illegal operation is not allowed and device operation is not guaranteed.

NOTE: The designation TS521X and TS511X refers to a portion of the part number designation of a series of commercial logic devices common in the industry. This number is normally preceded by a series of manufacturer specific characters to make up a complete part designation.

2 Device Standard

2.1 Description

The TS521X and TS511X device incorporate thermal sensing capability which is controlled and read over two wire bus. TS5 device operate from a nominal 1.8 V nominal power supply (VDDSPD) and a 1.0 V nominal power supply (VDDIO). The TS5 device is intended to operate up to 12.5 MHz on a 1.0V I3C Basic bus or up to 1 MHz on a 1.0 V to 3.3 V (Grade dependent) I^2C bus. The TS5 devices are intended to interface to I^2C or I3C Basic buses which have multiple devices on a shared bus, and must be uniquely addressed with fixed addressing on the same bus. All TS5 devices respond to specific pre-defined device select codes on the two-wire bus.

2.2 Common Features Summary

Table 1 — Device Part Numbers and Feature Summary

Part Number	Serial Numbers	Thermal Sensor	Temperature Range	Bus Pullup Nominal Voltage Range	Device Grade
TS5111	No	Yes	-40 °C to 125 °C	1.0 V to 3.3 V	A
TS5110	No			1.0 V to 1.2 V	B
TS5211	Yes	Yes	-40 °C to 125 °C	1.0 V to 3.3 V	A
TS5210	Yes			1.0 V to 1.2 V	B

2.2 Common Features Summary (cont'd)

- Two wire bus serial interface (I^2C and I3C Basic operation modes)
- Up to 12.5 MHz transfer rate
- 1.8 V and1.0 V power supply input
- Packaged in 6 ball WLCSP
- Packet Error Check (PEC) Function
- Parity Error Check Function
- Bus Reset Function
- Two unique addresses selected by SA pin
- In Band Interrupt (IBI) (Transparent Mode of Operation)
- 6 ball WLCSP Package device Functional Diagram

3 Device Power Up

The TS5 device has one VDDSPD supply input and one VDDIO supply input.

In order to prevent inadvertent operations during power up, a Power On Reset (POR) circuit is included. On cold power on, VDDSPD input supply must rise monotonically between VPON and $VDDSPD_{min}$ without ringback for device to turn on.

The TS5 device uses VDDIO input for its IO levels and it must reach VDDIOmin to ensure proper operation of I^2C or I3C Basic interface.

Once the VDDSPD and VDDIO supply is valid and stable, the TS5 device shall:

1. Within t_{Sense_SA} time, sense its SA pin to automatically configure the LID code based on what is detected on SA pin.
2. Enable I^2C interface within t_{INIT} time and be ready to receive the command from the controller. The TS5 device is ready for operation after t_{INIT} time.

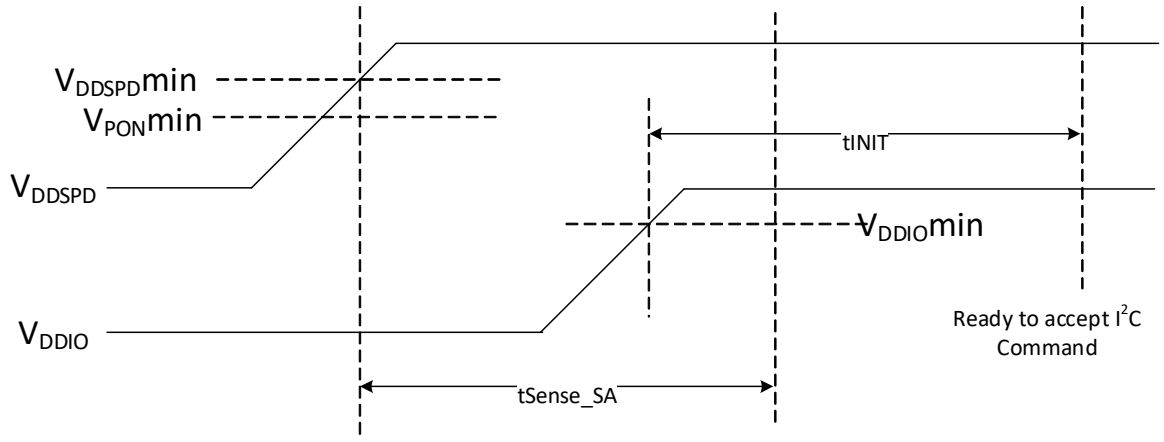


Figure 1 — Device Power Up Sequence

4 Device Reset and Initialization

At power down (phase during which VDDSPD input supply decreases continuously), as soon as VDDSPD input supply drops below the $VDDSPD_{min}$, the device does not guarantee the operation.

On warm power cycling, the VDDSPD and VDDIO input supply must remain below V_{POFF} for t_{POFF} and must meet cold power on reset timing when restoring the power.

5 I^2C and I3C Basic Operation

At power on, by default, the TS5 device comes up in I^2C mode of operation. Following applies in I^2C mode:

1. The maximum operation speed is limited to 1 MHz
2. In-band interrupts are not supported
3. Bus reset is supported.
4. Parity check is not supported except for supported CCCs.
5. Packet Error check is not supported.

The TS5 device shall operate in the I^2C mode until put into I3C Basic mode via command.

The controller may put the TS5 device in I3C Basic mode by issuing SETAASA CCC.

Following applies in I3C mode.

1. The maximum operation speed is up to 12.5 MHz
2. In-band interrupts are supported
3. Bus reset is supported.
4. Parity check is always enabled by default.
5. Packet error check is supported and by default is disabled.

6 Device Interface - Protocol

The 7-bit serial address of the TS5 device applies to both I^2C and I3C Basic mode of operation identically.

6.1 Serial Address of TS5 Device

The TS5 device 4-bit binary value is 0010b or 0110b depending sample status of SA pin on power up.

The TS5 device samples the status of the SA pin on power up. The sampled status of the SA pin is used to select one of the two possible unique LID code for the device. The selected LID code either 0010b or 0110b is merged with a 3 bit HID code Table 76, “MR7” [3:1] to establish the 7-bit address code the device. With the default setting in Table 76, “MR7” [3:1] = ‘111’; if the SA ball is connected to VSS, the device address shall be 0010 111b and if the SA ball is connected to VDDSPD, the device address shall be 0110 111b.

6.1 Serial Address of TS5 Device (cont'd)

Table 2 — 7-bit Address of TS5 Device

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	SA	1	0	1	1	1	R/W
TS5 Device Type ID (LID)				Host ID (HID)			Read/Write

6.2 Switch from I2C Mode to I3C Basic Mode

By default when TS5 first powers on, it operates in I2C mode. The TS5 device shall operate in I2C mode until put into I3C Basic mode via command.

In I2C mode, the controller is allowed to issued only 3 CCCs (DEVCTRL, SETHID, SETAASA). All other CCCs are not supported and the TS5 device may simply ignore it. The Controller must issue DEVCTRL and SETHID CCC first (if required) followed by SETAASA CCC.

The Controller puts the TS5 device in I3C mode by issuing SETAASA CCC.

When SETAASA CCC is registered by the TS5 device, it updates the Table 77, "MR18" [5] to '1'.

When SETHID CCC is registered by the TS5 device, it updates the Table 76, "MR7" [3:1].

6.3 Switch from I3C Basic Mode to I2C Mode

The Controller can put the TS5 device back in I2C mode from I3C Basic mode at any time by issuing RSTDAA CCC.

When RSTDAA CCC is registered by the TS5 device, it updates the Table 77, "MR18" [5] to '0'.

6.4 I2C Target Protocol

The TS5 device operate on a standard I2C serial interface. Transactions where the TS5 device is the targeted target device begin with the Controller issuing a START condition followed by a 7-bit TS5 device address then a read or write bit, RW. All data are transmitted with the most significant bit MSB first. During the address followed by R/W bit transmission, the TS5 device typically replies with an ACK unless there are exceptional conditions when it may passively assert a NACK.

The TS5 device accepts 1 byte of address which covers 256 bytes of registers. The TS5 device volatile register space does not require page selection process as all registers are within first 256 bytes.

6.4.1 Write Operation - Data Packet

Table 3 — Write Command Data Packet

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N	Stop
S or Sr[1]	0	X	1	0	HID			W=0	A	
	Address [7:0]								A	
	Data								A	
	...								A	
	Data								A	Sr or P

NOTE 1 In I2C mode, Start or Repeat Start operation followed by 7'h7E with W=0 is only allowed for the purpose of issuing CCCs that are allowed in I2C mode. Any other operation including another Repeat Start is considered an illegal operation.

6.4.2 Read Operation - Data Packet

Table 4 — Read Command Data Packet

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N	Stop
S or Sr[1]	0	X	1	0	HID			W=0	A	
	Address [7:0]								A	
Sr	0	X	1	0	HID			R=1	A[2]	
	Data								A	
	...								A	
	Data								N	Sr or P

NOTE 1 In I2C mode, Start or Repeat Start operation followed by 7'h7E with W=0 is only allowed for the purpose of issuing CCCs that are allowed in I2C mode. Any other operation including another Repeat Start is considered an illegal operation.

NOTE 2 If target device NACKs during Repeat Start for any reason, the controller my re-try Repeat Start again. The controller can do the Repeat Start as many times it may desire. TS5 may eventually ACK.

6.4.3 Default Read Address Pointer Mode

During normal operation of the DDR5 DIMM, the controller periodically may poll critical information from the same location. An example may be the TS device's temperature readout. To help improve the efficiency of the I2C bus protocol, the TS5 offers a default read address pointer mode so that whenever the TS5 device sees the STOP operation on its SCL and SDA bus, its read address pointer is always resets to default address. The default read pointer address mode is enabled through register Table 77, "MR18" [4] and default starting address for read operation is selectable through register Table 77, "MR18" [3:2]. This allows controller to read the read command data packet as shown in Table 5. The default read address pointer reduces the packet overhead by 2 bytes. The controller typically enables this mode when the normal operation of the DDR5 DIMM begins.

6.4.3 Default Read Address Pointer Mode (cont'd)

Table 5 — Read Command Data Packet with Default Address Pointer Mode

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N	Stop
S or Sr	0	X	1	0	HID			R=1	A	
	Data								A	
	...								A	
	Data								N	Sr or P

6.5 I3C Basic Target Protocol

The TS5 device operate on a standard I3C serial interface. Transactions where the TS5 device is the targeted target device begin with the Controller issuing a START condition followed by a 7-bit TS5 device address then a read or write bit, RW. All data are transmitted with the most significant bit MSB first. During the address followed by R/W bit transmission, the TS5 device typically replies with an ACK unless there are exceptional conditions when it may passively assert a NACK. See Table 6. The "T" bit carries Parity information from the Controller for each byte.

The Packet Error Code (PEC) function is disabled by default when the TS5 device is put in I3C Basic mode. The controller may optionally enable this function through Table 77, "MR18" [7] or DEVCTRL CCC. If enabled, the PEC is appended at the end of all transactions. If PEC is enabled, the controller must complete the burst length as indicated in CMD field. In other words, the controller must not interrupt the burst length pre-maturely for Write operation.

6.5.1 Write Operation - Data Packet

Table 6 — Write Command Data Packet; PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	0	X	1	0	HID			W=0	A[1,2,3]	
	Address [7:0]								T	
	Data								T	
	...								T	
	Data								T	Sr[4] or P

NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation (1st bit of Addr, bit [7]).

NOTE 2 The TS5 NACKs if there is a parity error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 3 The TS5 does not check for parity error in subsequent bytes when it determines the 7-bit device select code issued by the controller does not match with its own device code. The TS5 ignores the entire packet until STOP or next Repeat Start operation.

NOTE 4 Repeat Start or Repeat Start with 7'h7E.

6.5.1 Write Operation - Data Packet (cont'd)

Table 7 — Write Command Data Packet; PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	0	X	1	0	HID			W=0	A[1,2,3]	
	Address [7:0]								T	
	CMD			W=0	0000				T	
	Data								T	
	...								T	
	Data								T	
	PEC								T	Sr[4] or P

NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation (1st bit of Addr, bit [7]).

NOTE 2 The TS5 NACKs if there is a parity or PEC error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 3 The TS5 does not check for parity or PEC error in subsequent bytes when it determines the 7-bit device select code issued by the controller does not match with its own device code. The TS5 ignores the entire packet until STOP or next Repeat Start operation.

NOTE 4 Repeat Start or Repeat Start with 7'h7E.

The controller may optionally allow TS5 device to request IBI. For this case, the transactions to the TS5 device begin with the I3C controller issuing a START condition followed by 7'h7E and then write bit. If TS5 device has a pending IBI, it transmits its 7-bit device select code followed by R=1. If TS5 device has no pending IBI, there is no action taken by TS5. Table 8 and Table 9 show the I3C Basic bus write command data packet with optional IBI header for PEC disabled and PEC enabled case respectively. Note that in Table 9, PEC calculation does not include IBI header byte (7'h7E followed by W=0).

Table 8 — Write Command Data Packet with IBI Header; PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1,3]	
Sr	0	X	1	0	HID			W=0	A[2,3,4]	
	Address [7:0]								T	
	Data								T	
	...								T	
	Data								T	Sr[5] or P

NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation (Repeat Start).

NOTE 2 See Figure 4 to see how the transition occurs from Controller Push Pull Operation to target Open Drain (ACK) and See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation (1st bit of Addr, bit [7]).

NOTE 3 The TS5 NACKs if there is a parity error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 4 The TS5 does not check for parity error in subsequent bytes when it determines the 7-bit device select code issued by the controller does not match with its own device code. The TS5 ignores the entire packet until STOP or next Repeat Start operation.

NOTE 5 Repeat Start or Repeat Start with 7'h7E.

6.5.1 Write Operation - Data Packet (cont'd)

Table 9 — Write Command Data Packet with IBI Header; PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1,3]	
Sr	0	X	1	0	HID			W=0	A[2,3,4]	
	Address [7:0]								T	
	CMD			W=0	0000				T	
	Data								T	
	...								T	
	Data								T	
	PEC								T	Sr[5] or P

NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation (Repeat Start).

NOTE 2 See Figure 4 to see how the transition occurs from Controller Push Pull Operation to target Open Drain (ACK) and See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation (1st bit of Addr, bit [7]).

NOTE 3 The TS5 NACKs if there is a parity or PEC error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 4 The TS5 does not check for parity or PEC error in subsequent bytes when it determines the 7-bit device select code issued by the controller does not match with its own device code. The TS5 ignores the entire packet until STOP or next Repeat Start operation.

NOTE 5 Repeat Start or Repeat Start with 7'h7E.

6.5.2 Read Operation - Data Packet

The TS5 device operate on a standard I3C Basic serial interface. Transactions where the TS5 device is the targeted target device begin with the Controller issuing a START condition followed by a 7-bit TS5 device address then a read or write bit, RW. All data are transmitted with the most significant bit MSB first. During the address followed by R/W bit transmission, the TS5 device typically replies with an ACK unless there are exceptional conditions when it may passively assert a NACK. See Table 10. The "T" bit carries Parity information from the Controller for each byte prior to Repeat START. After Repeat START, "T" bit carries information from TS5 device to Controller indicating Continuous ('1') or Stop ('0') whether it is transmitting the last byte or not.

The Packet Error Code (PEC) function is disabled by default when the TS5 device is put in I3C Basic mode. The controller may optionally enable this function through Table 77, "MR18" [7] or DEVCTRL CCC. If enabled, the PEC is appended as shown in Table 10. If PEC is enabled, the controller must complete the burst length as indicated in CMD field. In other words, the controller must not interrupt the burst length pre-maturely for Read operation

6.5.2 Read Operation - Data Packet (cont'd)

Table 10 — Read Command Data Packet; PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	0	X	1	0	HID			W=0	A[1,2,3]	
	Address [7:0]								T	
Sr	0	X	1	0	HID			R=1	A/N[4,5]	
	Data								T=1	
	...								T=1	
	Data								T=1[6,7]	Sr[8] or P

NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation (1st bit of Addr, bit [7]).

NOTE 2 The TS5 NACKs if there is a parity error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 3 The TS5 does not check for parity error in subsequent bytes when it determines the 7-bit device select code issued by the controller does not match with its own device code. The TS5 ignores the entire packet until STOP or next Repeat Start operation.

NOTE 4 If target device NACKs during Repeat Start for any reason, the controller may re-try Repeat Start again. The controller can do the Repeat Start as many times it may desire. If target device NACKs due to parity error in previous bytes, it will always NACK regardless of how many times the Controller tries Repeat Start. If there were no parity errors, the TS5 may eventually ACK.

NOTE 5 See Figure 4 to see how the transition occurs from Controller Push Pull Operation to target Open Drain (ACK).

NOTE 6 See Figure 5 to see how Controller ends target device operation.

NOTE 7 When last byte (i.e., MR255) is reached (extreme rare case), the target device sends T = '0'. See Figure 6 to see how target device ends the operation followed by Controller STOP operation.

NOTE 8 Repeat Start or Repeat Start with 7'h7E.

6.5.2 Read Operation - Data Packet (cont'd)

Table 11 — Read Command Data Packet; PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	0	X	1	0	HID			W=0	A[1,2,3]	
	Address [7:0]								T	
	CMD			R=1	0000				T	
	PEC								T	
Sr	0	X	1	0	HID			R=1	A/N[4,5]	
	Data								T=1	
	...								T=1	
	Data								T=1	
	PEC								T=0[6]	Sr[7] or P

NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation (1st bit of Addr, bit [7]).

NOTE 2 The TS5 NACKs if there is a parity or PEC error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 3 The TS5 does not check for parity or PEC error in subsequent bytes when it determines the 7-bit device select code issued by the controller does not match with its own device code. The TS5 ignores the entire packet until STOP or next Repeat Start operation.

NOTE 4 If target device NACKs during Repeat Start for any reason, the controller may re-try Repeat Start again. The controller can do the Repeat Start as many times it may desire. If target device NACKs due to PEC error or parity error in previous bytes, it will always NACK regardless of how many times the Controller tries Repeat Start. If there were no parity or PEC errors, the TS5 may eventually ACK. The PEC calculation by the target device only includes device select code of the ACK response of the Repeat start operation. In other words, if there are more than one Repeat Start operation, the target device includes device select of only the last Repeat Start from the Controller when it ACKs in PEC calculation and all other NACK responses of the device select code of the Repeat Start are not included in PEC calculation.

NOTE 5 See Figure 4 to see how the transition occurs from Controller Push Pull Operation to target Open Drain (ACK).

NOTE 6 See Figure 6 to see how target device ends the operation followed by Controller STOP operation.

NOTE 7 Repeat Start or Repeat Start with 7'h7E.

The controller may optionally allow TS5 device to request IBI. For this case, the transactions to the TS5 device begin with the I3C Basic controller issuing a START condition followed by 7'h7E and then write bit. If TS5 device has a pending IBI, it transmits its 7-bit device select code followed by R=1. If TS5 device has no pending IBI, there is no action taken by TS5. Table 8 and Table 9 show the I3C Basic bus write command data packet with optional IBI header for PEC disabled and PEC enabled case respectively. Note that in Table 9, PEC calculation does not include IBI header byte (7'h7E followed by W=0).

6.5.2 Read Operation - Data Packet (cont'd)

Table 12 — Read Command Data Packet with IBI Header; PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1,3]	
Sr	0	X	1	0	HID			W=0	A[2,3,4]	
	Address [7:0]								T	
Sr	0	X	1	0	HID			R=1	A/N[5,6]	
	Data								T=1	
	...								T=1	
	Data								T=1[7,8]	Sr[9] or P

NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation (Repeat Start)

NOTE 2 See Figure 4 to see how the transition occurs from Controller Push Pull Operation to target Open Drain (ACK) and see Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation (1st bit of Addr, bit [7]).

NOTE 3 The TS5 NACKs if there is a parity error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 4 The TS5 does not check for parity error in subsequent bytes when it determines the 7-bit device select code issued by the controller does not match with its own device code. The TS5 ignores the entire packet until STOP or next Repeat Start operation.

NOTE 5 See Figure 4 to see how the transition occurs from Controller Push Pull Operation to target Open Drain (ACK).

NOTE 6 If target device NACKs during Repeat Start for any reason, the controller may re-try Repeat Start again. The controller can do the Repeat Start as many times it may desire. If target device NACKs due to parity error in previous bytes, it will always NACK regardless of how many times the Controller tries Repeat Start. If there were no parity errors, TS5 may eventually ACK.

NOTE 7 See Figure 5 to see how Controller ends target device operation.

NOTE 8 When last byte (i.e., MR255) is reached (extreme rare case), the target device sends T = '0'. See Figure 6 to see how target device ends the operation followed by Controller STOP operation.

NOTE 9 Repeat Start or Repeat Start with 7'h7E.

6.5.2 Read Operation - Data Packet (cont'd)

Table 13 — Read Command Data Packet with IBI Header; PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1,3]	
Sr	0	X	1	0	HID			W=0	A[2,3,4]	
	Address [7:0]								T	
	CMD			R=1	0000				T	
	PEC								T	
Sr	0	X	1	0	HID			R=1	A/N[5,6]	
	Data								T=1	
	...								T=1	
	Data								T=1	
	PEC								T=0[7]	Sr[8] or P

NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation (Repeat Start).

NOTE 2 See Figure 4 to see how the transition occurs from Controller Push Pull Operation to target Open Drain (ACK) and see Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation (1st bit of Addr, bit [7]).

NOTE 3 The TS5 NACKs if there is a parity or PEC error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 4 The TS5 does not check for parity or PEC error in subsequent bytes when it determines the 7-bit device select code issued by the controller does not match with its own device code. The TS5 ignores the entire packet until STOP or next Repeat Start operation.

NOTE 5 See Figure 4 to see how the transition occurs from Controller Push Pull Operation to target Open Drain (ACK).

NOTE 6 If target device NACKs during Repeat Start for any reason, the controller may re-try Repeat Start again. The controller can do the Repeat Start as many times it may desire. If target device NACKs due to PEC error or parity error in previous bytes, it will always NACK regardless of how many times the Controller tries Repeat Start. If there were no parity or PEC errors, the TS5 may eventually ACK. The PEC calculation by the target device only includes device select code of the ACK response of the Repeat start operation. In other words, if there are more than one Repeat Start operation, the target device includes device select of only the last Repeat Start from the Controller when it ACKs in PEC calculation and all other NACK responses of the device select code of the Repeat Start are not included in PEC calculation.

NOTE 7 See Figure 6 to see how target device ends the operation followed by Controller STOP operation.

NOTE 8 Repeat Start or Repeat Start with 7'h7E.

6.5.3 Default Read Address Pointer Mode

This mode works the same exact way as explained in Section 6.4.3. Table 14 and Table 15 show the read command data packet for PEC function disabled and enabled respectively. When PEC function is enabled, Table 77, "MR18" [1] sets the number of bytes that TS5 device sends out followed by the PEC calculation. If PEC is enabled, the controller must complete the burst length as indicated in Table 77, "MR18" [1] register. In other words, the controller must not interrupt the burst length pre-maturely for default address pointer read operation.

6.5.3 Default Read Address Pointer Mode (cont'd)

Table 14 — Read Command Data Packet with Read Address Pointer Mode; PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	0	X	1	0	HID			R=1	A[1]	
	Data								T=1	
	...								T=1	
	Data								T=1[2,3]	Sr[4] or P

NOTE 1 The TS5 NACKs if there is a parity error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 2 See Figure 5 to see how Controller ends target device operation.

NOTE 3 When last byte (i.e., MR255) is reached (extreme rare case), the target device sends T = '0'. See Figure 6 to see how target device ends the operation followed by Controller STOP operation.

NOTE 4 Repeat Start or Repeat Start with 7'h7E.

Table 15 — Read Command Data Packet with Read Address Pointer Mode; PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	0	X	1	0	HID			R=1	A[1]	
	Data								T=1	
	...								T=1	
	Data								T=1	
	PEC								T=0[2]	Sr[3] or P

NOTE 1 The TS5 NACKs if there is a parity or PEC error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 2 See Figure 6 to see how target device ends the operation followed by STOP operation

NOTE 3 Repeat Start or Repeat Start with 7'h7E.

6.5.3 Default Read Address Pointer Mode (cont'd)

Table 16 — Read Command Data Packet with Read Address Pointer and IBI Header; PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1,2]	
Sr	0	X	1	0	HID			R=1	A/N[2,3]	
	Data								T=1	
	...								T=1	
	Data								T=1[4,5]	Sr[6] or P

NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation (Repeat Start).

NOTE 2 The TS5 NACKs if there is a parity error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 3 See Figure 4 to see how the transition occurs from Controller Push Pull Operation to target Open Drain (ACK).

NOTE 4 See Figure 5 to see how Controller ends target device operation.

NOTE 5 When last byte (i.e., MR255) is reached (extreme rare case), the target device sends T = '0'. See Figure 6 to see how target device ends the operation followed by Controller STOP operation.

NOTE 6 Repeat Start or Repeat Start with 7'h7E.

Table 17 — Read Command Data Packet with Read Address Pointer and IBI Header; PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1,2]	
Sr	0	X	1	0	HID			R=1	A/N[2,3]	
	Data								T=1	
	...								T=1	
	Data								T=1	
	PEC								T=0[4]	Sr[5] or P

NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation (Repeat Start).

NOTE 2 The TS5 NACKs if there is a parity or PEC error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 3 See Figure 4 to see how the transition occurs from Controller Push Pull Operation to target Open Drain (ACK).

NOTE 4 See Figure 6 to see how target device ends the operation followed by STOP operation

NOTE 5 Repeat Start or Repeat Start with 7'h7E.

6.5.3 Default Read Address Pointer Mode (cont'd)

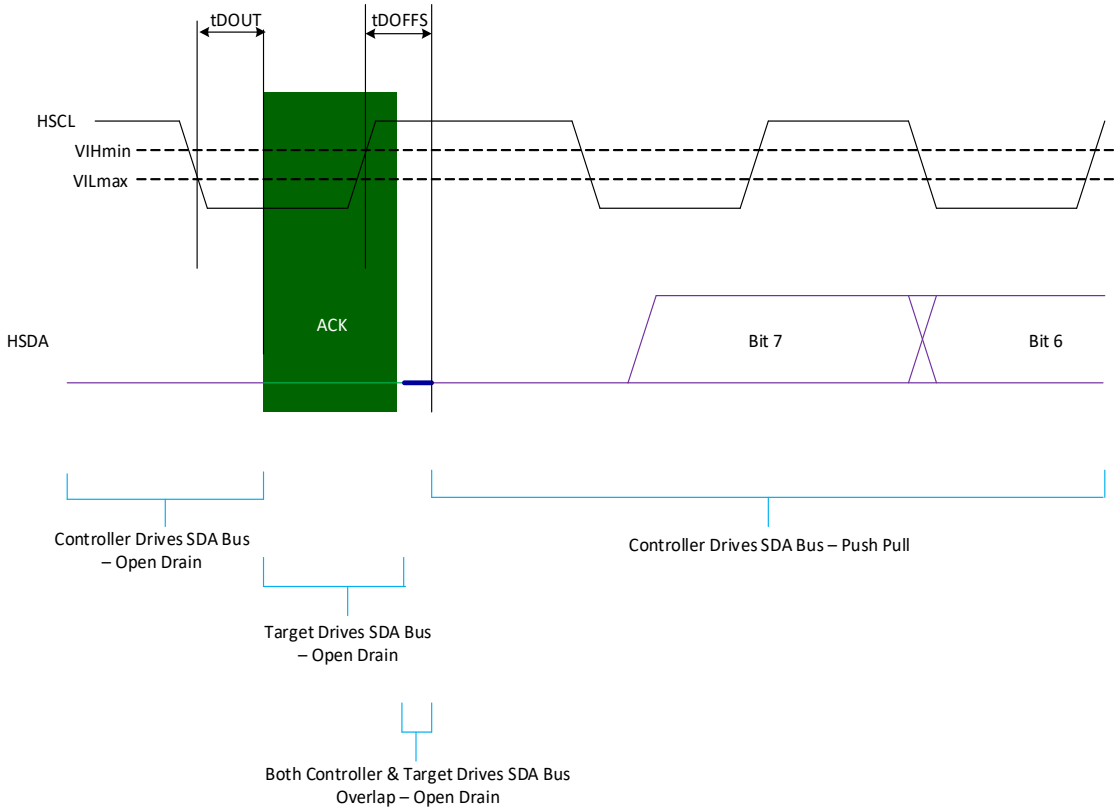


Figure 2 — Target Open Drain to Controller Push Pull Hand Off Operation

6.5.3 Default Read Address Pointer Mode (cont'd)

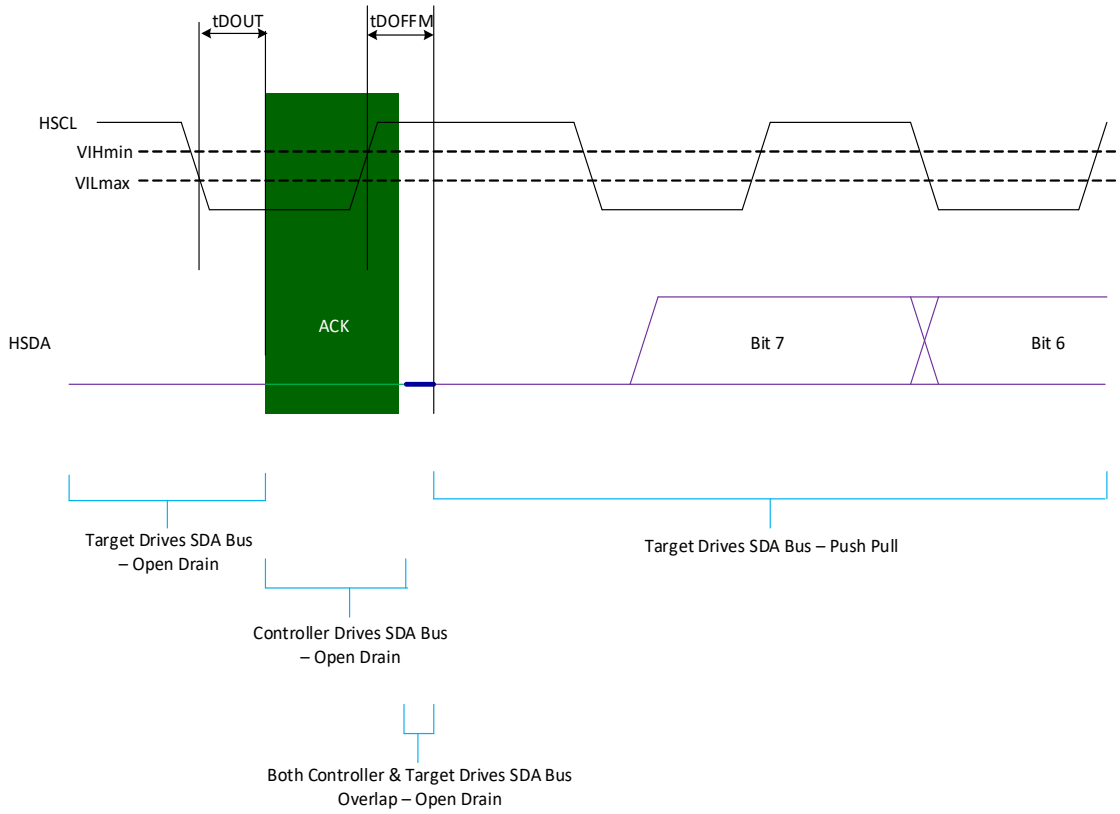


Figure 3 — Controller Open Drain (ACK) to Target Push Pull Hand Off Operation

6.5.3 Default Read Address Pointer Mode (cont'd)

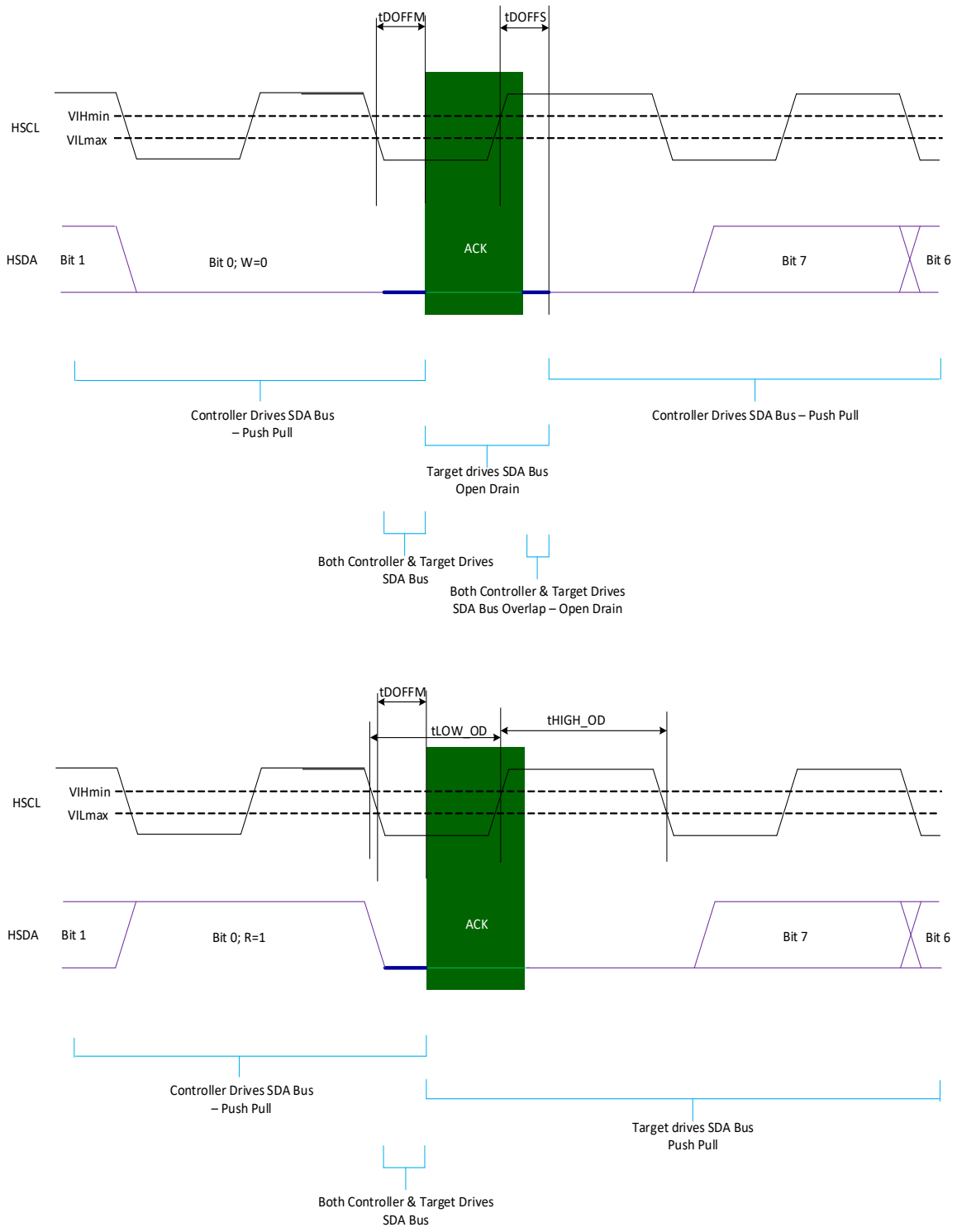


Figure 4 — Controller Push Pull to Target Open Drain Hand Off Operation

6.5.3 Default Read Address Pointer Mode (cont'd)

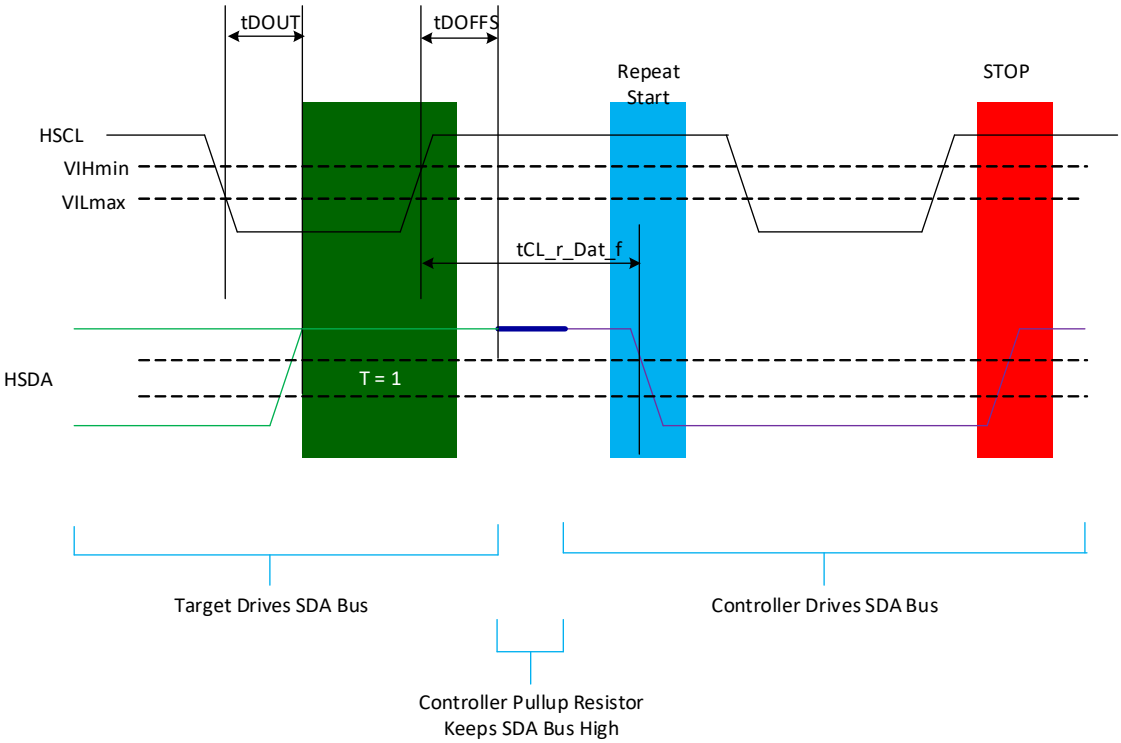


Figure 5 — T = 1; Controller Ends Read with Repeated START and STOP Waveform

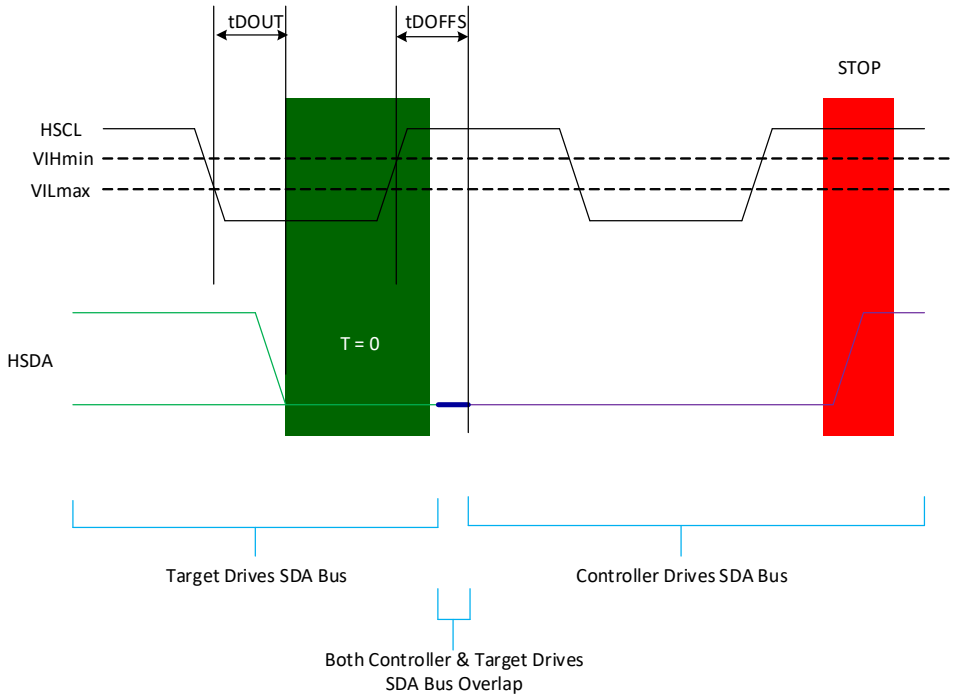


Figure 6 — T = 0; Target Ends Read; Controller Generates STOP

6.6 In Band Interrupt (IBI)

In I^2C mode, in band interrupt function is not supported. Only I3C Basic mode supports in band interrupt function.

6.6.1 Enabling and Disabling In Band Interrupt Function

By default, all interrupt sources are disabled (i.e., set to '0'). The controller may enable following interrupts in the TS5 device. Once enabled, the TS5 device sends an IBI when that event occurs.

1. Error Interrupt Enable in Table 81, "MR27" [4]:
   a. When Table 81, "MR27" [4] = '1', the device sends the IBI at next available opportunity when any of the register bit in Table 94, "MR52" [1:0] is set to '1' and sets Table 90, "MR48" [7] = '1' and updates Pending Interrupt Bits [3:0] = '0001' for GETSTATUS CCC.
   b. When Table 81, "MR27" [4] = '0', the device does not send the IBI regardless of the register bit status in Table 94, "MR52" [1:0]. However, the device does set Table 90, "MR48" [7] = '1' and updates Pending Interrupt Bits [3:0] = '0001' for GETSTATUS CCC.
2. Temp Sensor Interrupt Enable in Table 81, "MR27" [3:0]: The controller can set any combination of register bits to '1'.
   a. When any of the register bits in Table 81, "MR27" [3:0] = '1' and if Table 81, "MR27" [4] = '1', the device sends the IBI at next available opportunity when the corresponding register bit in Table 93, "MR51" [3:0] is set to '1' and sets Table 90, "MR48" [7] = '1' and updates Pending Interrupt Bits [3:0] = '0001' for GETSTATUS CCC.
   b. When any of the register bits in Table 81, "MR27" [3:0] = '0' or Table 81, "MR27" [4] = '0', the device does not send the interrupt regardless of the corresponding register bit status in Table 93, "MR51" [3:0]. However, the device does set Table 90, "MR48" [7] = '1' and updates Pending Interrupt Bits [3:0] = '0001' for GETSTATUS CCC if any of the bits in Table 81, "MR27" [3:0] = '1' and Table 81, "MR27" [4] = '0'.

6.6.2 Mechanics of Interrupt Generation

Event interrupts may be generated by the local device if IBI is enabled. When there is a pending interrupt (i.e., Table 90, "MR48" [7] = '1') and if IBI is enabled (i.e., Table 81, "MR27" [4] = '1') the TS5 device requests an interrupt after detecting START condition by transmitting its 7-bit binary address (LID bits followed by HID bits) followed by R/W = '1' on the SDA bus serially (synchronized by SCL falling transitions).

If TS5 device detects no START condition but if the I3C bus (SDA and SCL) has been inactive (no edges seen) for t_{AVAL} period, then TS5 device may assert SDA low by t_{IBI_ISSUE} time to request an interrupt. When the TS5 device requests an interrupt, the Controller toggles the SCL. The TS5 device transmits its 7-bit binary address (LID bits followed by HID bits) followed by R/W bit = '1' to the Controller.

When the TS5 device requests an interrupt, the controller may take one of the two actions below.

- The Controller sends ACK on 9^{th} bit to accept the interrupt request. At this point, if the TS5 device confirms that it has won the arbitration, the TS5 device transmits the IBI payload as shown in Table 18 and Table 19 for PEC disabled and PEC enabled configuration respectively. See Figure 7. Figure 7 just shows only first two data bits of the MDB byte to illustrate the timing. The interrupt payload contains MDB followed by Table 93, "MR51" and Table 94, "MR52" bytes in order. The controller then issues the STOP command. Note the timing waveform in Figure 7. The controller then accepts the IBI payload if it sends an ACK on 9^{th} bit to accept the interrupt request. The controller can interrupt the IBI payload at T bit.

6.6.2 Mechanics of Interrupt Generation (cont'd)

If controller stops the IBI payload at T bit in the middle of payload, the TS5 retains the IBI status flag (Table 90, "MR48" [7]) and Pending Interrupt Bits [3:0] internally and waits for the next opportunity to request an interrupt. If the TS5 device successfully transmits the entire IBI payload, it then clears IBI status flag and Pending Interrupt Bits [3:0] = '0000' on its own and does not request for an IBI again unless there is another different event occurs; for another same event, the device does not request for an IBI.

- The Controller sends NACK on the 9th bit as shown in Figure 8 followed by a STOP command. In this case, the TS5 device does not transmit the IBI payload and waits for the next opportunity to request an interrupt. At this point, though Controller sent an NACK, it does have a knowledge of which TS5 device sent the IBI request. The TS5 device retains the IBI status flag and Pending Interrupt Bits [3:0] = '0001'

Table 18 — Target Device IBI Payload Packet; PEC is Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/T	Stop
S	0	X	1	0	HID			R=1	A[1]	
	MDB = 0x00								T=1	
	MR51 [7:0]								T=1	
	MR52 [7:0]								T=0[2]	P
NOTE 1	See Figure 3 to see how the transition occurs from Controller Open Drain (ACK) to target Push Pull Operation (1st bit of MDB Byte bit [7]).									
NOTE 2	See Figure 6 to see how target device ends the operation followed by Controller STOP operation.									

Table 19 — Target Device IBI Payload Packet; PEC is Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/T	Stop
S	0	X	1	0	HID			R=1	A[1]	
	MDB 0x00								T=1	
	MR51 [7:0]								T=1	
	MR52 [7:0]								T=1	
	PEC								T=0[2]	P
NOTE 1	See Figure 3 to see how the transition occurs from Controller Open Drain (ACK) to target Push Pull Operation (1st bit of MDB Byte bit [7]).									
NOTE 2	See Figure 6 to see how target device ends the operation followed by Controller STOP operation.									

6.6.2 Mechanics of Interrupt Generation (cont'd)

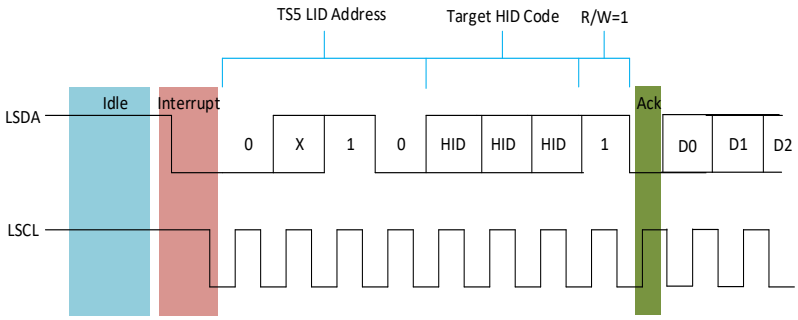


Figure 7 — TS5 Requests Interrupt, Controller Ack followed by TS5 Device IBI Payload

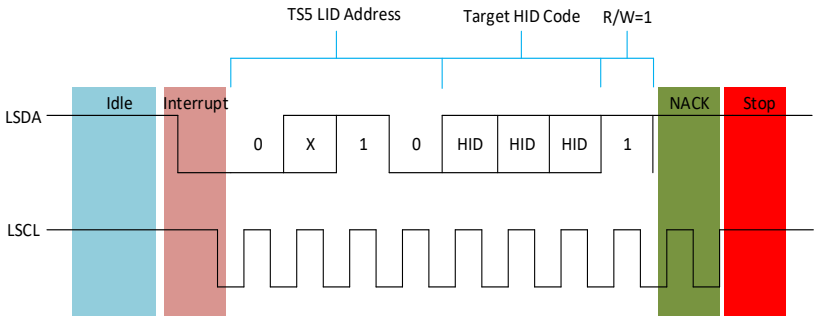


Figure 8 — TS5 Requests Interrupt; Controller NACK followed by STOP

6.6.3 Interrupt Arbitration

As there are multiple devices I3C Basic bus, multiple device may request an interrupt when the Controller I3C Basic bus is inactive for t_{AVAL} period. Arbitration process is required.

For DDR5 DIMM application environment, there could be up to total of 13 difference devices including the TS5 on I3C bus.

On a typical DDR5 DIMM application environment, all devices have the same 3-bit HID code. Hence the arbitration is always won by the lowest 4-bit LID code. For example, if one TS5 target device has LID code of '0010' and other device (PMIC) has a LID code of '1001', through the arbitration process, the LID code of TS5 '0010' wins. The other device (PMIC) with a LID code of '1001' must release the bus and wait for next opportunity to request an interrupt. Table 20 shows the arbitration priority based on the LID code for all devices. The Green color cells in Table 20 are the likely devices that will be on a standard DDR5 RDIMM or DDR5 LRDIMM. The Olive color cells in Table 20 do not apply.

6.6.3 Interrupt Arbitration (cont'd)

Table 20 — Interrupt Arbitration - Among All Devices

Device	LID Code	HID Code = '111'	Arbitration Priority
N/A	0000	N/A	N/A
RFU	0001	111	1
TS0	0010	111	2
RFU	0011	111	3
RFU	0100	111	4
RFU	0101	111	5
TS1	0110	111	6
RFU	0111	111	7
PMIC1	1000	111	8
PMIC0	1001	111	9
SPD Hub	1010	N/A	N/A
RCD	1011	111	10
PMIC2	1100	111	11
RFU	1101	111	12
RFU	1110	111	13
N/A	1111	N/A	N/A

In an uncommon but possible scenario would be that at the exact same time as when the Hub or local target devices (i.e., TS5) are requesting an interrupt, the controller is starting an operation to the Hub or local target devices (i.e., TS5). When this happens, Controller also gets involved in the arbitration process along with the Hub or the local target devices (TS5). During the arbitration phase, there will be always only one winning device and it could be either Hub or the local target device (i.e., TS5) or the Controller.

If the controller wins during the arbitration phase, it continues with normal operation. The losing Hub or local target device (i.e., TS5) waits for next opportunity to send an interrupt.

If the controller is loose during the arbitration phase, the controller must let go of the bus. When Controller looses during the arbitration, the controller must let the Hub or local target device (i.e., TS5) finish sending their 4-bit LID code followed by 3-bit HID code followed by R/W = '1'. At this point, during the 9^{th} bit, the controller has two options to take the action as noted below:

- Controller sends an ACK to accept the interrupt and hence accepts the IBI payload from the winning Hub or local target device (i.e., TS5). After the IBI payload, the controller issues STOP operation.
- Controller sends an NACK followed by STOP operation.

A rare but still possible scenario would be that at the exact same time as when the TS5 is requesting an interrupt, the controller is starting an operation to the same TS5. When this happens, neither Controller or nor the TS5 knows it is a winner until the 8^{th} bit and Controller always wins. This is because, the TS5 sends R=1 (8^{th} bit) during the interrupt. The controller sets W=0 (8^{th} bit) during the operation. As a result, the controller wins and the TS5 must let go of the bus and wait for the next opportunity to send an interrupt.

6.6.3 Interrupt Arbitration (cont'd)

An extremely rare but still possible scenario would be that at the same exact time as when TS5 device is requesting an interrupt, the controller is requesting a read operation with the default read address pointer mode to the TS5 device. When this happens, there is no winning device. This is the only time there is no winning device. This is because, the TS5 device sends R=1 (8^{th} bit) during the interrupt and Controller also sends R=1 for read request with default read address pointer mode. As a result, there is no winner because Controller is waiting for TS5 to ACK and TS5 is waiting for Controller to ACK. In this case, neither Controller nor TS5 will ACK. Since there is no ACK (i.e., NACK) by either device, the Controller must time out and repeat the read request with Repeat Start. When Controller repeats the read request with Repeat Start, the TS does not send an interrupt because of Repeat Start.

6.6.4 Clearing Device Status and IBI Status Registers

The TS5 device provides the IBI status in Table 90, "MR48" [7] by setting it to '1'. The TS5 device clears the IBI status register Table 90, "MR48" [7] to '0' automatically when it sends a complete IBI (including payload and without interruption) and it also clears Pending Interrupt Bits [3:0] to '0000'. Once IBI status register is cleared, the TS5 does not request for an IBI again unless an another event occurs.

The TS5 device provides the device status in Table 93, "MR51" and Table 94, "MR52" registers. The status information in Table 93, "MR51" and Table 94, "MR52" are latched and remains set even after the TS5 device sends IBI payload and clears the IBI status register Table 90, "MR48" [7] to '0'. The controller must explicitly clear the status register through Clear command by writing '1' for appropriate status or by issuing a Global clear command.

After Controller issues clear command, if the condition is no longer present, the TS5 device clears the appropriate status register, clears the IBI status register to '0' and Pending Interrupt Bits [3:0] to '0000' even if the TS5 device has not sent the IBI. After Controller issues clear command, if the condition is still present, the device will again set the appropriate status register, sets the IBI status register to '1' and Pending Interrupt Bits [3:0] to '0001' even if the device has already sent the IBI and entire IBI payload.

6.7 Packet Error Check (PEC) Function

In I^2C mode, packet error checking is not supported. Only I3C Basic mode supports packet error checking.

The TS5 device implement an 8-bit Packet Error Code (PEC) which is appended at the end of all transactions if PECs is enabled through DEVCTRL CCC or by directly writing '1' to Table 77, "MR18" [7]. The PEC is a CRC-8 value calculated on all the messages bytes except for START, STOP, REPEATED START conditions or T-bits, ACK and NACK and IBI header (7'h7E followed W=0) bits.

The polynomial for CRC-8 calculations is:

- $C(X) = X^8 + X^2 + X^1 + 1$

The seed value for PEC function is all zero.

When Controller calculates PEC for TS5 device, it includes LID and HID bits followed by R/W bit.

6.8 Parity Error Check Function

In I^2C mode, parity error checking is not supported except for supported CCCs. Only I3C mode supports parity error checking.

By default, when TS5 device is put in I3C mode, parity function is automatically enabled. The controller can disable the function after it is enabled. Controller can also disable the parity function with DEVCTRL CCC or by directly writing '1' to Table 77, "MR18" [6].

6.8 Parity Error Check Function (cont'd)

When parity function is disabled, the TS5 device simply ignores the "T" bit information from the Controller. The controller may actually choose to compute the parity and send that information during "T" bit or simply drive static low or high in "T" bit.

The TS5 device implements ODD parity. If an odd number of bits in the byte are '1', the parity bit value is '0'. If even number of bits in the byte are '1', the parity bit value is '1'. The controller computes the parity and sends during "T" bit.

6.9 Packet Error Check and Parity Error Handling

There are two types of error checking done by the TS5 device. Parity error checking and Packet Error checking. By default, the parity error checking is always enabled and packet error checking is disabled. The controller may enable the packet error checking at any time. The parity error is checked for each byte in a packet except for the device select code byte from the controller. The controller sends parity error information in "T" bit.

I3C basic defines TE0, TE1, TE2, TE3, TE4, TE5, TE6 error detection for target devices. Only TE1 and TE2 error detection is supported by the TS5 for parity checking. All other errors are not supported and not applicable.

6.9.1 Write Command Data Packet Error Handling - PEC Disabled

The TS5 device checks for the parity error for each byte in a packet that it receives from the controller except for the device select code byte that it receives from the controller.

Table 21 — Write Command Data Packet; PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	0	X	1	0	HID			W=0	A[1,2,3]	
	Address [7:0]								T	
	Data								T	
	...								T	
	Data								T	Sr[4] or P

NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation (1st bit of Addr, bit [7]).

NOTE 2 The TS5 NACKs if there is a parity error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 3 The TS5 does not check for parity error in subsequent bytes when it determines the 7-bit device select code issued by the controller does not match with its own device code. The TS5 ignores the entire packet until STOP or next Repeat Start operation.

NOTE 4 Repeat Start or Repeat Start with 7'h7E.

Write command - if no parity error:

- The TS5 device executes the command.

Write command - if parity error:

- The TS5 device discards the byte in the packet that had a parity error.
- The TS5 device discards all subsequent bytes in that packet until the STOP operation. The TS5 device may or may not check parity for all sub-sequent bytes in that packet.

6.9.1 Write Command Data Packet Error Handling - PEC Disabled (cont'd)

- Note that as the packet contains more than one byte, if first byte had no parity error but the second byte had a parity error, the TS5 device may or may not execute the first byte operation but second byte and all subsequent bytes operations are discarded.
- The TS5 device sets the Table 94, "MR52" [0] and Table 90, "MR48" [7] and P_Err in GETSTATUS CCC to '1'; updates Pending Interrupt Bits [3:0] in GETSTATUS CCC to '0001' and waits for the next opportunity to send an in band interrupt if IBI is enabled.

6.9.2 Read Command Data Packet Error Handling - PEC Disabled

The TS5 device checks for parity error for each byte in a packet except for the device select code byte that it receives from the controller prior to Repeat Start as shown in Table 22.

The TS5 device does not compute the parity when it sends the data to the Controller. The does not check for parity error for the bytes shown in Table 22. The device sends Continuous ('1') or Stop ('0') information during "T" bit when TS5 device is sending the read data.

Table 22 — Read Command Data Packet; PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	0	X	1	0	HID			W=0	A[1,2,3]	
	Address [7:0]								T	
Sr	0	X	1	0	HID			R=1	A/N[4,5]	
	Data								T=1	
	...								T=1	
	Data								T=1[6,7]	Sr[8] or P

NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation (1st bit of Addr, bit [7]).

NOTE 2 The TS5 NACKs if there is a parity error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 3 The TS5 does not check for parity error in subsequent bytes when it determines the 7-bit device select code issued by the controller does not match with its own device code. The TS5 ignores the entire packet until STOP or next Repeat Start operation.

NOTE 4 If target device NACKs during Repeat Start for any reason, the controller may re-try Repeat Start again. The controller can do the Repeat Start as many times it may desire. If target device NACKs due to parity error in previous bytes, it will always NACK regardless of how many times the Controller tries Repeat Start. If there were no parity errors, TS5 may eventually ACK.

NOTE 5 See Figure 4 to see how the transition occurs from Controller Push Pull Operation to target Open Drain (ACK).

NOTE 6 See Figure 5 to see how Controller ends target device operation.

NOTE 7 When last byte (i.e., MR255) is reached (extreme rare case), the target device sends T = '0'. See Figure 6 to see how target device ends the operation followed by Controller STOP operation.

NOTE 8 Repeat Start or Repeat Start with 7'h7E.

Read Command - If no parity error:

- The TS5 sends ACK back to the controller when Controller performs Start Repeat operation.
- The TS5 device executes the command and sends the data as shown in Table 22.

6.9.2 Read Command Data Packet Error Handling - PEC Disabled (cont'd)

Read Command - If parity error:

- The TS5 device discards the byte in the packet that had a parity error.
- The TS5 device sends NACK back to the controller when Controller performs a Start Repeat operation. This is shown in the RED color cell in Table 22 above. The NACK represents either a parity error or that TS5 is not able to start the read operation. The controller may re-try Repeat Start again. The controller may do the Repeat Start as many times as it may desire. If the TS5 target device NACKs due to parity error in a previous byte from the controller, it will always NACK regardless of how many times Controller tries Repeat Start.
- The TS5 does not send the data shown in Table 22 and instead expects Controller to perform STOP operation.
- The TS5 device sets Table 94, "MR52" [0] and Table 90, "MR48" [7], and P_Err in GETSTATUS CCC to '1'; updates Pending Interrupt Bits [3:0] in GETSTATUS CCC to '0001' and waits for the next opportunity to send an in band interrupt if IBI is enabled.

6.9.3 Write Command Data Packet Error Handling - PEC Is Enabled

The TS5 device checks for the parity error for each byte in a packet that it receives from the controller except for the device select code byte that it receives from the controller as shown in Table 23. Further, the TS5 device checks for the packet error for the entire packet (from Start condition until last byte of Data) that it receives from the controller as shown in Table 23.

Table 23 — Write Command Data Packet; PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	0	X	1	0	HID			W=0	A[1,2,3]	
	Address [7:0]								T	
	CMD			W=0	0	0	0	0	T	
	Data								T	
	...								T	
	Data								T	
	PEC								T	Sr[4] or P

NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation (1st bit of Addr, bit [7]).

NOTE 2 The TS5 NACKs if there is a parity or PEC error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 3 The TS5 does not check for parity or PEC error in subsequent bytes when it determines the 7-bit device select code issued by the controller does not match with its own device code. The TS5 ignores the entire packet until STOP or next Repeat Start operation.

NOTE 4 Repeat Start or Repeat Start with 7'h7E.

Write command - if no parity error:

- The TS5 device waits for the entire packet. If no error in packet, the TS5 device executes the command. If there is an error in the packet, the TS5 device discards the entire packet and does not execute that packet and waits for STOP; sets the Table 94, "MR52" [1] and Table 90, "MR48" [7] and PEC_Err in GETSTATUS CCC to '1';updates Pending Interrupt Bits [3:0] in GETSTATUS CCC to '0001' and waits for the next opportunity to send in band interrupt if IBI is enabled.

6.9.3 Write Command Data Packet Error Handling - PEC Is Enabled (cont'd)

Write command - if parity error:

- The TS5 device discards that byte and the entire packet until STOP operation.
- The TS5 device sets Table 94, "MR52" [0] and Table 90, "MR48" [7] and P_Err in GETSTATUS CCC to '1'; updates Pending Interrupt Bits [3:0] in GETSTATUS CCC to '0001' and waits for the next opportunity to send in band interrupt if IBI is enabled.
- The TS5 device may or may not check the error for the packet. If the TS5 device checks for the packet error, likely it will detect an error in the packet and the device may also set Table 94, "MR52" [1] and PEC_Err in GETSTATUS CCC to '1' as well.

6.9.4 Read Command Data Packet Error Handling - PEC Is Enabled

The TS5 device checks for parity error for each byte in a packet except for the device select code byte that it receives from the controller prior to Repeat Start as shown in Table 24.

The TS5 device does not compute the parity when it sends the data to the Controller. The does not check for parity error for the bytes shown in Table 24. The device sends Continuous ('1') or Stop ('0') information during "T" bit when TS5 device is sending the read data.

The TS5 device checks for the PEC error in a packet that it receives from Controller from Start condition to Repeat Start (from first device select code followed by the address offset and CMD byte).

The TS5 device computes the packet error code for the entire packet starting with Repeat Start (device select code and the data TS5 device transmits back to Controller).

6.9.4 Read Command Data Packet Error Handling - PEC Is Enabled (cont'd)

Table 24 — Read Command Data Packet; PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	0	X	1	0	HID			W=0	A[1,2,3]	
	Address [7:0]								T	
	CMD			R=1	0	0	0	0	T	
	PEC								T	
Sr	0	X	1	0	HID			R=1	A/N[4,5]	
	Data								T=1	
	...								T=1	
	Data								T=1	
	PEC								T=0[6]	Sr[7] or P

NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation (1st bit of Addr, bit [7]).

NOTE 2 The TS5 NACKs if there is a parity or PEC error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 3 The TS5 does not check for parity or PEC error in subsequent bytes when it determines the 7-bit device select code issued by the controller does not match with its own device code. The TS5 ignores the entire packet until STOP or next Repeat Start operation.

NOTE 4 If target device NACKs during Repeat Start for any reason, the controller my re-try Repeat Start again. The controller can do the Repeat Start as many times it may desire. If target device NACKs due to PEC error or parity error in previous bytes, it will always NACK regardless of how many times the Controller tries Repeat Start. If there were no parity or PEC errors, TS5 may eventually ACK. The PEC calculation by the target device only includes device select code of the ACK response of the Repeat start operation. In other words, if there are more than one Repeat Start operation, the target device includes device select of only the last Repeat Start from the Controller when it ACKs in PEC calculation and all other NACK responses of the device select code of the Repeat Start are not included in PEC calculation.

NOTE 5 See Figure 4 to see how the transition occurs from Controller Push Pull Operation to target Open Drain (ACK).

NOTE 6 See Figure 6 to see how target device ends the operation followed by Controller STOP operation.

NOTE 7 Repeat Start or Repeat Start with 7'h7E.

Read command - If no parity error and no PEC error

- The TS5 device sends ACK back to the controller when Controller performs a Start Repeat operation.
- The TS5 device executes the command and sends the data as shown in Table 24.
- The TS5 computes PEC for the bytes (from Start condition to PEC byte prior to Repeat Start) shown in the cells in Table 24.

Read command - if parity error or PEC error

- The TS5 device discards the byte in the packet that had a parity error.
- The TS5 device discards second byte in that packet if a parity error occurred in first byte. The TS5 device may or may not check parity for the second byte in that packet.
- The TS5 device discards the packet if there is a PEC error.
- The TS5 sends NACK back to the controller when Controller performs Start Repeat operation. This is shown in the RED color cell in Table 24 above. The NACK represents either PEC error or a parity error in one of the three bytes or that TS5 is not able to start the read operation.

6.9.4 Read Command Data Packet Error Handling - PEC Is Enabled (cont'd)

The controller may re-try Repeat Start again. The controller may do the Repeat Start as many times it may desire. The PEC calculation by TS5 device only includes device select code of the ACK responses of the Repeat Start operation. In other words, if there are more than one Repeat Start operation, the TS5 device includes the device select of only the last Repeat Start from the Controller when it ACKs in PEC calculation and other NACK responses of the device select codes of the Repeat Start are not included in PEC calculation. If the TS5 target device NACKs due to PEC error or a parity error in a previous bytes from Controller, it will always NACK regardless of how many times Controller tries Repeat Start.

- The TS5 device does not send any data shown in Table 24 and instead expects Controller to perform STOP operation.
- The TS5 device sets Table 94, "MR52" [0] and Table 90, "MR48" [7] and P_Err in GETSTATUS CCC to '1' for parity error and Table 94, "MR52" [1] and Table 90, "MR48" [7] and PEC_Err in GETSTATUS CCC to '1' for PEC error. Further, TS5 updates Pending Interrupt Bits [3:0] in GETSTATUS CCC to '0001' and waits for the next opportunity to send an in band interrupt if IBI is enabled.

6.10 CCC Packet Error Handling

Parity error and PEC error detected in a CCC packet are handled the same way as described for normal Read/Write operations.

6.11 Error Reporting

All error conditions detected by the TS5 devices are captured in Table 93, "MR51" and Table 94, "MR52" registers.

There are three different possible ways error information can be communicated to the controller.

1. The controller makes the read request to Table 93, "MR51" and Table 94, "MR52" registers.
2. The controller starts any transactions with 7'h7E IBI header (Only applicable in I3C mode).
3. The TS5 device sends in band interrupt if enabled, when its SCL and SDA input has been idle for t_{AVAL} time (only applicable in I3C Basic mode).

6.12 I3C Basic Common Command Codes (CCC)

The I3C Basic spec lists large number of Common Command Codes (CCC). Not all CCC are required to be supported. The TS5 device NACKs for all unsupported CCC. The TS5 supports CCC as listed in Table 25.

The TS5 device requires STOP operation in between when switching from CCC operation to private device specific Write or Read or Default Read Address Pointer mode operation and vice versa. In other words, any CCC operation must be followed by STOP operation before continuing to any device specific Write or Read or Default Read Address Pointer mode operation. Similarly, any device specific Write or Read or Default Read Address Pointer mode operation must be followed by STOP operation before continuing to any CCC operation. The TS5 device also requires STOP operation from any direct CCC to broadcast CCC.

The TS5 device does allow Repeat Start operation from between any direct CCC to any other direct CCC or between any broadcast CCC to any other broadcast CCC or between any private Write or Read or Default Read Address Pointer mode operation to any other private Write or Read or Default Read Address Pointer mode operation.

6.12 I3C Basic Common Command Codes (CCC) (cont'd)

Table 25 — TS5 CCC Support Requirement

CCC	Mode	Code	Description	Note
ENEC	Broadcast	0x00	Enable Event Interrupts	
	Direct	0x80		
DISEC	Broadcast	0x01	Disable Event Interrupts	
	Direct	0x81		
RSTDAA	Broadcast	0x06	Put the device in I^2C Mode (aka: Reset Dynamic Address Assignment)	
SETAASA	Broadcast	0x29	Put the device in I3C Basic Mode (aka: Set All Addresses to Static Address)	
GETSTATUS	Direct	0x90	Get Device Status	
DEVCAP	Direct	0xE0	Get Device Capability	1
SETHID	Broadcast	0x61	TS5 updates 3-bit HID field	1
DEVCTRL	Broadcast	0x62	Configure SPD Hub and all devices behind Hub	1
NOTE 1 JEDEC specific CCC.				

6.12.1 ENEC CCC

The ENEC CCC is only supported after device is put in I3C Basic mode. In I^2C mode, it is illegal for controller to issue this CCC. When ENEC CCC is registered by the TS5, it updates Table 81, "MR27" [4] = '1' and it takes in effect at the next Start operation (i.e., after STOP condition). Table 26 to Table 29 show an example of a single ENEC CCC. Table 30 shows the encoding definition for ENEC CCC.

If PEC function is enabled, the PEC calculation starts with Start or Repeat Start operation but does not include 7'h7E with W=0 byte in PEC calculation.

Table 26 — ENEC CCC - Broadcast

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x00 (Broadcast)								T	
	0x00							ENINT	T	Sr[2] or P
NOTE 1 The TS5 NACKs if there is a parity error in a previous transaction when controller performs consecutive transactions with Repeat Start. NOTE 2 Repeat Start or Repeat Start with 7'h7E.										

6.12.1 ENEC CCC (cont'd)

Table 27 — ENEC CCC - Broadcast with PEC

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x00 (Broadcast)								T	
	0x00							ENINT	T	
	PEC								T	Sr[2] or P

NOTE 1 The TS5 NACKs if there is a parity or PEC error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 2 Repeat Start or Repeat Start with 7'h7E.

Table 28 — ENEC CCC - Direct

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x80 (Direct)								T	
Sr	DevID[6:0]							W=0	A	
	0x00							ENINT	T	Sr[2] or P

NOTE 1 The TS5 NACKs if there is a parity error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 2 Repeat Start or Repeat Start with 7'h7E.

Table 29 — ENEC CCC - Direct with PEC

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x80 (Direct)								T	
	PEC								T	
Sr	DevID[6:0]							W=0	A	
	0x00							ENINT	T	
	PEC								T	Sr[2] or P

NOTE 1 The TS5 NACKs if there is a parity or PEC error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 2 Repeat Start or Repeat Start with 7'h7E.

6.12.1 ENEC CCC (cont'd)

Table 30 — ENEC CCC Byte Encoding

Bit	Encoding	Notes
ENINT	0 = No Action 1 = Enable IBI Interrupt	It is illegal for Controller to issue ENEC CCC with ENINT bit = '0'

6.12.2 DISEC CCC

The DISEC CCC is only supported after device is put in I3C Basic mode. In I^2C mode, it is illegal for controller to issue this CCC. When DISEC CCC is registered by the TS5, it updates Table 81, "MR27" [4] = '0' and it takes in effect at the next Start operation (i.e., after STOP condition). Table 31 to Table 34 show an example of a single DISEC CCC. Table 35 shows the encoding definition for DISEC CCC.

If PEC function is enabled, the PEC calculation starts with Start or Repeat Start operation but does not include 7'h7E with W=0 byte in PEC calculation.

Table 31 — DISEC CCC - Broadcast

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x01 (Broadcast)								T	
	7'h00							DISINT	T	Sr[2] or P

NOTE 1 The TS5 NACKs if there is a parity error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 2 Repeat Start or Repeat Start with 7'h7E.

Table 32 — DISEC CCC - Broadcast with PEC

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x01 (Broadcast)								T	
	7'h00							DISINT	T	
	PEC								T	Sr[2] or P

NOTE 1 The TS5 NACKs if there is a parity or PEC error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 2 Repeat Start or Repeat Start with 7'h7E.

6.12.2 DISEC CCC (cont'd)

Table 33 — DISEC CCC - Direct

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x81 (Direct)								T	
Sr	DevID[6:0]							W=0	A	
	0x00							DISINT	T	Sr[2] or P

NOTE 1 The TS5 NACKs if there is a parity error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 2 Repeat Start or Repeat Start with 7'h7E.

Table 34 — DISEC CCC - Direct with PEC

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x81 (Direct)								T	
	PEC								T	
Sr	DevID[6:0]							W=0	A	
	0x00							DISINT	T	
	PEC								T	Sr[2] or P

NOTE 1 The TS5 NACKs if there is a parity or PEC error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 2 Repeat Start or Repeat Start with 7'h7E.

Table 35 — DISEC CCC Byte Encoding

Bit	Encoding	Notes
DISINT	0 = No Action 1 = Disable IBI Interrupt	It is illegal for Controller to issue DISEC CCC with DISINT bit = '0'

6.12.3 RSTDAA CCC

The RSTDAA CCC is only supported after device is put in I3C Basic mode. In I^2C mode, this CCC is ignored. When RSTDAA CCC is registered by the TS5, it updates Table 77, "MR18" [5] = '0' and it takes in effect at the next Start operation (i.e., after STOP condition). Further it disables IBI and PEC function (Table 81, "MR27" [4] = '0', Table 77, "MR18" [7] = '0', respectively) and clears parity function Table 77, "MR18" [6] = '0').

Table 36 to Table 37 show an example of a single RSTDAA CCC.

If PEC function is enabled, the PEC calculation starts with Start or Repeat Start operation but does not include 7'h7E with W=0 byte in PEC calculation.with W=0 byte in PEC calculation.

6.12.3 RSTDAA CCC (cont'd)

Table 36 — RSTDAA CCC - Broadcast

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x06 (Broadcast)								T	Sr[2] or P

NOTE 1 The TS5 NACKs if there is a parity error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 2 Repeat Start or Repeat Start with 7'h7E.

Table 37 — RSTDAA CCC - Broadcast with PEC

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x06 (Broadcast)								T	
	PEC								T	Sr[2] or P

NOTE 1 The TS5 NACKs if there is a parity or PEC error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 2 Repeat Start or Repeat Start with 7'h7E.

6.12.4 SETAASA CCC

The SETAASA CCC is only supported when device is in I^2C mode. In I^2C mode, when controller issues CCC, to guarantee that this CCC is registered by the device without any error, the controller shall limit the maximum speed operation for this CCC to 1 MHz. In I3C Basic mode, this CCC is ignored. When SETAASA CCC is registered by the TS5, it updates Table 77, "MR18" [5] = '1' and it takes in effect at the next Start operation (i.e., after STOP condition). Table 38 shows an example of a single SETAASA CCC.

SETAASA CCC does not support PEC function as device is in I^2C mode and there is no PEC function in I^2C mode.

Table 38 — SETAASA CCC - Broadcast

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A	
	0x29 (Broadcast)								T	P

6.12.5 GETSTATUS CCC

The GETSTATUS CCC is supported in I3C Basic mode. In I^2C mode, this CCC is ignored (i.e., it is not executed internally and GETSTATUS CCC code is not acknowledged and controller must do STOP operation). Table 39 to Table 40 show an example of a single GETSTATUS CCC.

If PEC function is enabled, the PEC calculation starts with Start or Repeat Start operation but does not include 7'h7E with W=0 byte in PEC calculation.

6.12.5 GETSTATUS CCC (cont'd)

Table 39 — GETSTATUS CCC - Direct

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x90 (Direct)								T	
Sr	DevID[6:0]							R=1	A	
	PEC_Err	0	0	0	0	0	0	0	T	
	0	0	P_Err	0	Pending Interrupt				T	Sr[2] or P

NOTE 1 The TS5 NACKs if there is a parity error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 2 Repeat Start or Repeat Start with 7'h7E.

Table 40 — GETSTATUS CCC - Direct with PEC[1]

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[2]	
	0x90 (Direct)								T	
	PEC								T	
Sr	DevID[6:0]							R=1	A	
	PEC_Err	0	0	0	0	0	0	0	T	
	0	0	P_Err	0	Pending Interrupt				T	
	PEC								T	Sr[3] or P

NOTE 1 GETSTATUS CCC with PEC check is only supported in I3C Basic mode

NOTE 2 The TS5 NACKs if there is a parity or PEC error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 3 Repeat Start or Repeat Start with 7'h7E.

Table 41 — GETSTATUS CCC Byte Encoding

Bit	Encoding	Notes
PEC_Err	0 = No Error 1 = PEC Error Occurred	This register is cleared when Controller issues clear command to Table 79, "MR20" [1] for PEC error
P_Err	0 = No Error 1 = Protocol Error; Parity Error occurred	This register is cleared when Controller issues clear command to Table 79, "MR20" [0] for Parity error.
Pending Interrupt	0000 = No Pending Interrupt 0001 = Pending Interrupt All other encodings are reserved	This register is cleared when Controller issues clear command to any appropriate device status register that causes IBI status register to get cleared.

6.12.5 GETSTATUS CCC (cont'd)

When the TS5 device responds to GETSTATUS CCC, after it completes the response, the PEC_Err, P_Err and Pending Interrupt Bits [3:0] do not automatically get cleared. The controller must explicitly clear the appropriate status register through Clear command by writing '1' to corresponding register or by issuing Global Clear command. Once the TS5 device clears the appropriate status register, only then PEC_Err, P_err and Pending Interrupt Bits [3:0] gets cleared.

After controller issues clear command, if the condition is still present, the device will again set the appropriate status register, sets the IBI status register to '1' and Pending Interrupt Bits [3:0] to '0001'.

6.12.6 DEVCAP CCC

The DEVCAP CCC is only supported after device is put in I3C Basic mode. In I^2C mode, it is illegal for controller to issue this CCC. Table 42 to Table 43 show an example of a single DEVCAP CCC. Table 44 defines the encoding for DEVCAP CCC.

If PEC function is enabled, the PEC calculation starts with Start or Repeat Start operation but does not include 7'h7E with W=0 byte in PEC calculation.

Table 42 — DEVCAP CCC - Direct

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0xE0 (Direct)								T	
Sr	DevID[6:0]							R=1	A[1]	
	MSB (Each bit defines capability)								T	
	LSB (Each bit defines capability)								T	Sr[2] or P

NOTE 1 The TS5 NACKs if there is a parity error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 2 Repeat Start or Repeat Start with 7'h7E.

Table 43 — DEVCAP CCC - Direct with PEC

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0xE0 (Direct)								T	
	PEC								T	
Sr	DevID[6:0]							R=1	A[1]	
	MSB (Each bit defines capability)								T	
	LSB (Each bit defines capability)								T	
	PEC								T	Sr[2] or P

NOTE 1 The TS5 NACKs if there is a parity or PEC error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 2 Repeat Start or Repeat Start with 7'h7E.

6.12.6 DEVCAP CCC (cont'd)

Table 44 — DEVCAP CCC Byte Encoding

Bit	Encoding	Notes
MSB [7]	RFU	Coded as '0'
MSB[6]	RFU	Coded as '0'
MSB[5]	RFU	Coded as '0'
MSB[4]	RFU	Coded as '0'
MSB[3]	RFU	Coded as '0'
MSB[2]	0 = No Support for Timer based Reset 1 = Supports Timer based Reset	Coded as '1'
MSB[1:0]	RFU	Coded as '000'
LSB[7:0]	RFU	Coded as '000'

6.12.7 SETHID CCC

The SETHID CCC is supported only when device is in I^2C mode. In I^2C mode, when controller issues CCC, to guarantee that this CCC is registered by the device without any error, the controller shall limit the maximum speed operation for this CCC to 1 MHz. In I3C Basic mode, it is illegal for controller to issue this CCC. When SETHID CCC is registered by the TS5, it updates Table 76, "MR7" [3:1] with the HID code received by the TS5 and it takes in effect at the next Start operation (i.e., after STOP condition). Table 45 shows an example of a single SETHID CCC. As the device is in I^2C mode when SETHID CCC is issued, the PEC function is not supported.

Once TS5 receives SETHID CCC and updates its 3-bit HID code, after the Stop operation, TS5 device only responds to updated 7-bit address. The 4-bit LID code of the TS5 device remains as is.

The Controller may issue SETHID CCC more than one time.

Table 45 — SETHID CCC - Broadcast

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A	
	0x61 (Broadcast)								T	
	0	0	0	0	HID[2:0]			0	T	P

6.12.8 DEVCTRL CCC

On a typical I3C Basic bus, there can be up to 120 devices. For DDR5 DIMM application environment, there are up to 8 SPD5 Hub devices and behind each SPD5 Hub devices, there are 4 local target devices totaling up to 40 or more devices on I3C Basic bus. For certain operation such as enable or disable functions that are common to all devices (i.e., Packet Error Check), the controller must go through one device at a time which takes significant amount of time at initial power up. Further, it requires additional complexity on the controller because it must speak different protocol depending on how it may access the device until all devices are configured identically.

6.12.8 DEVCTRL CCC (cont'd)

To help expedite this configuration operation and to simplify the controller complexity, the device supports the DEVCTRL CCC. The DEVCTRL CCC is supported either in I^2C mode or I3C Basic mode of operation. In I^2C mode, when controller issues CCC, to guarantee that this CCC is registered by the device without any error, the controller shall limit the maximum speed operation for this CCC to 1 MHz. Table 46 to Table 47 show an example of a single DEVCTRL CCC.

If PEC function is enabled, the PEC calculation starts with Start or Repeat Start operation but does not include 7'h7E with W=0 byte in PEC calculation.

The controller shall pay attention to DEVCTRL CCC. If DEVCTRL CCC is used to access device specific registers (e.g. RegMod = '1'), the controller shall still follow any device specific register restriction. For example, if device specific register require STOP operation for device to take in the effect of the setting, the controller must also use STOP operation when using DEVCTRL CCC to access device specific register.

Table 46 — DEVCTRL CCC - Broadcast

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x62 (Broadcast)								T	
	AddrMask[2:0]			StartOffset[1:0]		PEC BL[1:0]		RegMod	T	
	DevID[6:0]							0	T[2]	
	Byte 0 Data Payload								T	
	Byte 1 Data Payload								T	
	Byte 2 Data Payload								T	
	Byte 3 Data Payload								T	Sr[3] or P

NOTE 1 The TS5 NACKs if there is a parity error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 2 An exception is made for DEVCTRL CCC. TS5 does report parity error when it determines 7-bit device select code issued by the controller does not match with its own device code. If 7-bit device select code does not match but if parity is still valid, the device does not check for parity error in subsequent bytes; ignores the entire packet and waits until STOP or Repeat Start operation.

NOTE 3 Repeat Start or Repeat Start with 7'h7E.

6.12.8 DEVCTRL CCC (cont'd)

Table 47 — DEVCTRL CCC - Broadcast with PEC[1]

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[2]	
	0x62 (Broadcast)								T	
	AddrMask[2:0]			StartOffset[1:0]		PEC BL[1:0]		RegMod	T	
	DevID[6:0]							0	T[3]	
	Byte 0 Data Payload								T	
	Byte 1 Data Payload								T	
	Byte 2 Data Payload								T	
	Byte 3 Data Payload								T	
	PEC								T	Sr[4] or P

NOTE 1 DEVCTRL CCC with PEC check is only supported in I3C Basic mode.

NOTE 2 The TS5 NACKs if there is a parity or PEC error in a previous transaction when controller performs consecutive transactions with Repeat Start.

NOTE 3 An exception is made for DEVCTRL CCC. TS5 does report parity error when it determines 7-bit device select code issued by the controller does not match with its own device code. The device does not check for PEC as all subsequent bytes are discarded due to parity error. If 7-bit device select code does not match but if parity is still valid, the device does not check for parity error in subsequent bytes; ignores the entire packet and waits until STOP or next Repeat Start operation.

NOTE 4 Repeat Start or Repeat Start with 7'h7E.

6.12.8 DEVCTRL CCC (cont'd)

Table 48 — DEVCTRL CCC Command Definition

Parameter	Definition
AddrMask[2:0]	Broadcast, Unicast or Multicast Command Selection 000 = Unicast Command; TS5 device responds if DevID[6:0] field matches with TS5 device's own 7-bit address (4-bit LID + 3-bit HID) 011 = Multicast Command; TS5 device and possible other device responds if DevID[6:3] field matches with TS5 device's own 4-bit LID address 111 = Broadcast Command; All devices responds to this command All other encodings are reserved
StartOffset[1:0]	Only applicable if RegMod = '0' Identifies the starting Byte (Byte 0 or Byte 1 or Byte 2 or Byte 3) for DEVCTRL CCC. Controller can start at any Byte (from Byte 0 to Byte 3) and has continuous access to next byte until STOP operation. If Byte 3 is reached, the controller is responsible for applying STOP operation. 00 = Byte 0 01 = Byte 1 10 = Byte 2 11 = Byte 3
PEC BL[1:0]	Only applicable if RegMod = '0' and PEC function is enabled. Identifies the burst length just for this DEVCTRL CCC. The device uses the setting in this field to know when the PEC byte is expected after the data bytes. 00 = 1 Byte 01 = 2 Byte 10 = 3 Byte 11 = 4 Byte
RegMod	Identifies if DEVCTRL is going to be used for General Registers as identified in Byte 0 to Byte 3 or device specific address offset register. 0 = Access to General Registers in Byte 0 to Byte 3 (i.e., StartOffset[1:0] = Valid) 1 = Device Specific Offset Address (i.e., StartOffset[1:0] and PECBL[1:0] is a don't care and does not apply). The Controller shall NOT use RegMod = '1' with Broadcast Command if there are different types of devices on the I3C Basic bus.
DevID[6:0]	Identifies 7-bit device address. Device responds to DEVCTRL CCC data packet depending on AddrMask[2:0]. If AddrMask[2:0] = '111', DevID[6:0] is a don't care and device always responds. If AddrMask[2:0] = '000', DevID[6:0] must match for device to respond If AddrMask[2:0] = '011', DevID[6:3] must match for device to respond. DevID[2:0] is don't care. For any other codes for AddrMask[2:0], the device always NACKs.

6.12.8 DEVCTRL CCC (cont'd)

Table 49 — DEVCTRL CCC Data Payload Definition

Byte #	Bit #	Function	Definition	Comment
Byte 0	[7]	PEC Enable	0 = Disable 1 = Enable	Table 77, "MR18" [7] is updated
	[6]	Parity Disable	0 = Enable 1 = Disable	Table 77, "MR18" [6] is updated
	[5:2]	RFU	RFU	
	[1]	RSVD	0 = RSVD 1 = RSVD	TS5 device always ignores this bit.
	[0]	RFU	RFU	
Byte 1	[7:4]	RFU	RFU	
	[3]	Global and IBI Clear	0 = No Action 1 = Clear All Event and pending IBI[1]	Table 81, "MR27" [7] is updated.
	[2:0]	RFU	RFU	
Byte 2	[7:0]	RFU	RFU	
Byte 3	[7:0]	RFU	RFU	

NOTE 1 After target device clears the event, the device can still have certain registers set to '1' if the event is still present, in which case, the device will generate an IBI again at the next opportunity.

6.12.8.1 DEVCTRL CCC Examples - RegMod = '0'

Table 50 shows an example of DEVCTRL CCC data packet. It assumes that all devices on the bus are already in I3C Basic mode with PEC function disabled and parity function enabled. In this example, the Controller uses DEVCTRL CCC as Multicast command. Controller sends Multicast command to all devices with 4-bit LID code of '1001' on I3C bus to do VR Enable followed by all devices with 4-bit LID code of '0110' to disable parity function. The controller sends AddrMask = '011' to indicate Multicast command with DevID[6:3] match; StartOffset = '00' to indicate starting Byte 0 and RegMod = '0' to indicates general register. Upon receiving this command, all devices with DevID[6:3] that matches to '1001' will do the VR Enable command and DevID[6:3] that matches to '0110' with disable the parity function.

6.12.8.1 DEVCTRL CCC Examples - RegMod = ‘0’ (cont’d)

Table 50 — DEVCTRL CCC Example - Multicast Command to ‘1001’ and ‘0110’ Devices

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1]	
	0x62 (Broadcast)								T	
	011			00		00		0	T	
	1001 000							0	T	
	0000 0010								T	
Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x62 (Broadcast)								T	
	011			00		00		0	T	
	0110 000							0	T	
	0100 0000								T	P

NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation.

Table 51 shows an example of DEVCTRL CCC data packet. It assumes that all devices on the bus are already in I3C Basic mode with PEC function disabled and parity function enabled. In this example, the Controller uses DEVCTRL CCC as Broadcast command to enable PEC function. The controller sends AddrMask = ‘111’ to indicate Broadcast command; StartOffset = ‘00’ to indicate starting Byte 0 and RegMod = ‘0’ to indicates general register. Upon receiving this command, all devices will enable PEC function.

Table 51 — DEVCTRL CCC Example - Broadcast Command to all Devices

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1]	
	0x62 (Broadcast)								T	
	111			00		00		0	T	
	0000 000							0	T	
	1000 0000								T	P

NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation.

Table 52 shows an example of DEVCTRL CCC data packet. It assumes that all devices on the bus are already in I3C Basic mode with PEC function disabled and parity function enabled. In this example, the Controller uses DEVCTRL CCC as Unitcast command to enable VR on DIMM5. The controller sends AddrMask = ‘000’ to indicate Unicast command; StartOffset = ‘00’ to indicate starting Byte 0 and RegMod = ‘0’ to indicates general register. Upon receiving this command, PMIC on DIMM5 will enable its regulator.

6.12.8.1 DEVCTRL CCC Examples - RegMod = ‘0’ (cont’d)

Table 52 — DEVCTRL CCC Example - Unicast Command to PMIC on DIMM5

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1]	
	0x62 (Broadcast)								T	
	000			00		00		0	T	
	1001 101							0	T	
	0000 0010								T	P

NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation.

6.12.8.2 DEVCTRL CCC Examples - RegMod = ‘1’

Table 53 shows an example of DEVCTRL CCC data packet for the purpose of configuring device specific address offset register. It assumes that all devices on the bus are already in I3C Basic mode with PEC function enabled and parity function enabled. In this example, the Controller sends Multicast command to all devices with 4-bit LID code of ‘0010’ on the I3C Basic bus to write to address offset of 0x1C and 0x1D with data 0xFF and 0x55, respectively, followed by all devices with 4-bit LID of ‘1001’ on the I3C bus to write to address offset of 0x15 with data 0x78.

The PEC calculation starts with Start or Repeat Start operation but does not include 7’h7E with W=0 byte in PEC calculation.

Table 53 — DEVCTRL CCC Example - Multicast Command to ‘0010’ and ‘1001’ Devices

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1]	
	0x62 (Broadcast)								T	
	011			00		00		1	T	
	0010 000							0	T	
	0001 1100 (address offset 0x1C)								T	
	0010 0000 (CMD field = 2 bytes of data)								T	
	1111 1111 (data)								T	
	0101 0101 (data)								T	
	PEC								T	
Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x62 (Broadcast)								T	
	011			00		00		1	T	
	1001 000							0	T	
	0001 0101 (address offset 0x15)								T	
	0000 0000 (CMD field = 1 byte of data)								T	
	0111 1000 (data)								T	
	PEC								T	P

NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation.

6.12.8.2 DEVCTRL CCC Examples - RegMod = ‘1’ (cont’d)

Table 54 shows an example of DEVCTRL CCC data packet for the purpose of configuring device specific address offset register. It assumes that all devices on the bus are already in I3C Basic mode with PEC function disabled and parity function enabled. In this example, the Controller sends Multicast command to all devices with 4-bit LID code of ‘1001’ on the I3C Basic bus to write to address offset of 0x13 with data 0xFF and it continues to write data 0x01 to the next address.

Table 54 — DEVCTRL CCC Example - Multicast Command to ‘1001’ Devices

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1]	
	0x62 (Broadcast)								T	
	011			00		00		1	T	
	1001 000							0	T	
	0001 0011 (address offset 0x13)								T	
	1111 1111 (data)								T	
	0000 0001 (data)								T	P
NOTE 1 See Figure 2 to see how the transition occurs from target Open Drain (ACK) to Controller Push Pull Operation.										

6.13 IO Operation

At power on, by default, the TS5 device comes up in legacy I^2C mode of operation with Open Drain IO for its interface. The maximum speed is limited to 1 MHz and supported IO voltage levels are from 1.0 V to 3.3 V for Grade A and 1.0 V to 1.2 V for Grade B.

After power on, the controller may put the TS5 device in I3C mode of operation. In I3C Basic mode of operation, the maximum speed is limited to 12.5 MHz and supported IO voltage levels are from 1.0 V to 1.2 V.

In I3C Basic mode, the controller may drive the SCL clock input of the TS5 device using either Push-Pull output driver or using the open-drain output driver. It is expected that for all DDR5 DIMM family environment, the controller may always drive the SCL clock input using a Push-Pull output driver.

To support in band interrupt, the TS5 device supports dynamic switching between Open Drain mode and Push Pull mode on its SDA bus for various event. The Table 55 below describes the different mode of operation by the TS5 device for each cycle.

6.13 IO Operation (cont'd)

Table 55 — TS5 Device Dynamic IO Operation Mode Switching

	Open Drain Mode	Push Pull Mode
START + Device Select Code	Yes	No
START + 7'h7E IBI Header Byte	Yes	No
REPEAT START + Device Select Code	No	Yes
REPEAT START + 7'h7E Header Byte	No	Yes
CCC Bytes (i.e., after 7'h7E+W=0+ACK)	No	Yes
STOP	No	Yes
ACK/NACK Responses	Yes	No
Command, Block Address, Address Operation	No	Yes
Interrupt Request by Target + Device Select Code	Yes	No
IBI Payload	No	Yes
Write Data, T-bit sequence	No	Yes
Read Data, T-bit sequence	No	Yes
PEC, T-bit sequence	No	Yes

6.14 Bus Clear

The TS device supports the following described Bus Clear feature in I^2C mode only. Any attempt by controller to perform I^2C Bus clear on a target device in I3C mode may result in an active drive bus contention on the SDA data line.

There may be abnormal circumstances when the controller abruptly stops clocking SCL while the target device is in the middle of outputting data for read operation. For these type of events, the SDA data line may appear as stuck low as the device is expecting to receive more clock pulses from the controller. Eventually when the controller has control of the SCL clock, the controller may optionally clear the device that is stuck low on the SDA data line by sending continuous 18 clock pulses without driving the SDA data line followed by STOP operation. The device floats the SDA line within 18 clock pulses and returns to the Idle state. The device is ready for normal new transaction with Start condition.

6.15 Bus Reset

To prevent a malfunctioning device from locking up the I^2C bus or I3C Basic bus, a bus reset mechanism is defined. It uses a timeout mechanism on SCL as shown in Figure 9 to force a device bus reset. All devices on a I^2C or I3C Basic bus reset simultaneously. Bus reset operation works same way regardless of whether device is operating in I^2C or I3C Basic mode.

To guarantee the device resets I^2C bus or I3C Basic bus, the SCL clock input Low time has to be greater than or equal to $t_{TIMEOUT(Max)}$.

The TS5 device does not reset I^2C bus or I3C Basic bus if the SCL clock input Low time is less than $t_{TIMEOUT(Min)}$.

If the SCL clock input Low time is between $t_{TIMEOUT(Min)}$ and $t_{TIMEOUT(Max)}$, the TS5 device does not guarantee and it may or may not reset the I^2C bus or I3C Basic bus.

6.15 Bus Reset (cont'd)

When RESET, the TS5 device takes following action.

1. Interface and any pending command or transactions are cleared
2. All internal register values are preserved unless noted otherwise in item # 3 below.
3. Device returns to I^2C mode of operation; Table 76, "MR7" [3:1] resets to '111'; Table 77, "MR18" [7:5] resets to '000'; Table 81, "MR27" [4] resets to '0'; and Table 94, "MR52" [1:0] resets to '00'.
4. Device does not re-sample SA pin.
5. Device floats the SDA pin such that it gets pulled High by external/other device pullup.
6. Device treats bus reset as STOP operation.

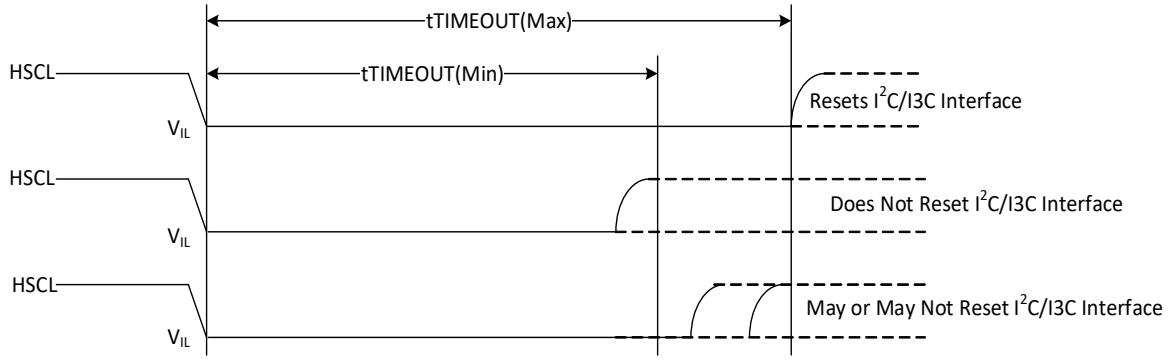


Figure 9 — I^2C or I3C Basic Bus Reset - TS5 Device

6.16 Command Truth Table

The command truth table as shown in Table 56 only applies in I3C mode with PEC enabled. In I^2C mode and I3C Basic mode with PEC disabled, the command truth table does not apply.

Table 56 — For I3C Basic Mode Only with PEC Enabled - Command Truth Table

TS5 Command	Command Name	CMD Code	RW	Address
		2nd Byte Bits [7:5]	2nd Byte Bit [4]	1st Byte Bits [7:0]
Write 1 Byte to Register	W1R	000	0	V
Read 1 Byte from Register	R1R		1	V
Write 2 Byte to Register	W2R	001	0	V
Read 2 Byte from Register	R2R		1	V
Reserved	RSVD	010 to 111	RSVD	RSVD

7 Device Pin Definition

7.1 TS5 Pin Definition

Table 57 — Pin Description - TS521X, TS5000

Pin #	Pin Name	Pin Type	Description
C2	VDDSPD	Power	1.8 V Input Power Supply. Connect minimum of 0.1 µF capacitor to VSS.
C1	VSS	GND	GND
A1	SCL	I	I2C or I3C Basic Input Clock
B1	SDA	IO	I2C or I3C Basic Data
B2	SA	I	I2C or I3C Basic Address Pin. This pin is tied to either VSS or VDDSPD to establish the 4-bit LID.
A2	VDDIO	Power	1.0 V Input Power Supply. Connect minimum of 0.1 µF capacitor to VSS

8 AC Timing Definition

8.1 I2C or I3C Basic Bus Timing

The TS5 device follows the I2C or I3C Basic bus timing requirements. Figure 10 and Figure 11 show the timing diagram for Data bus Input and Data Output parameters.

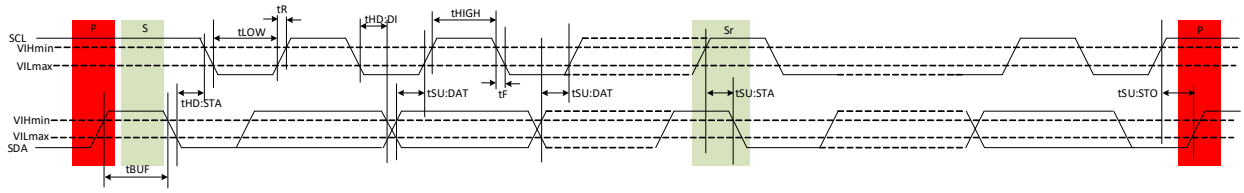


Figure 10 — I2C or I3C Basic Bus AC Input Timing Parameter Definition

8.1 I²C or I3C Basic Bus Timing (cont'd)

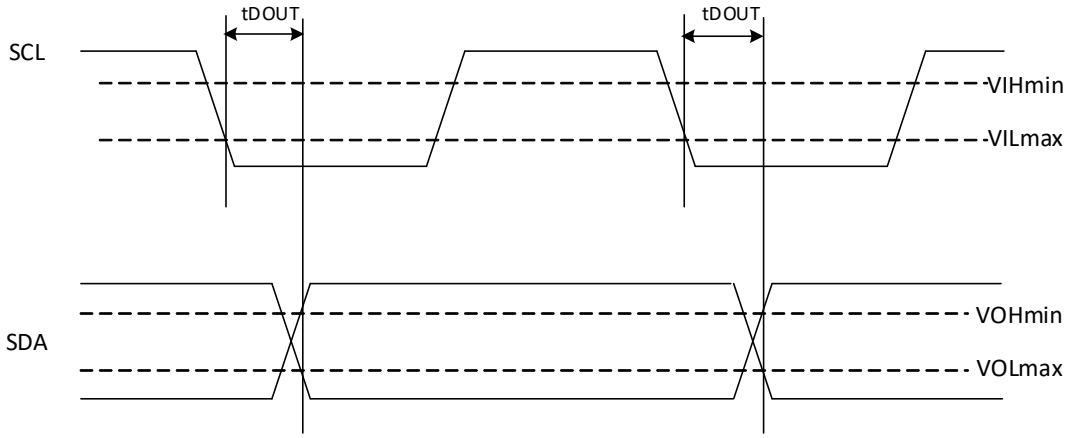


Figure 11 — I3C Basic Bus AC Data Output Timing Parameter Definition

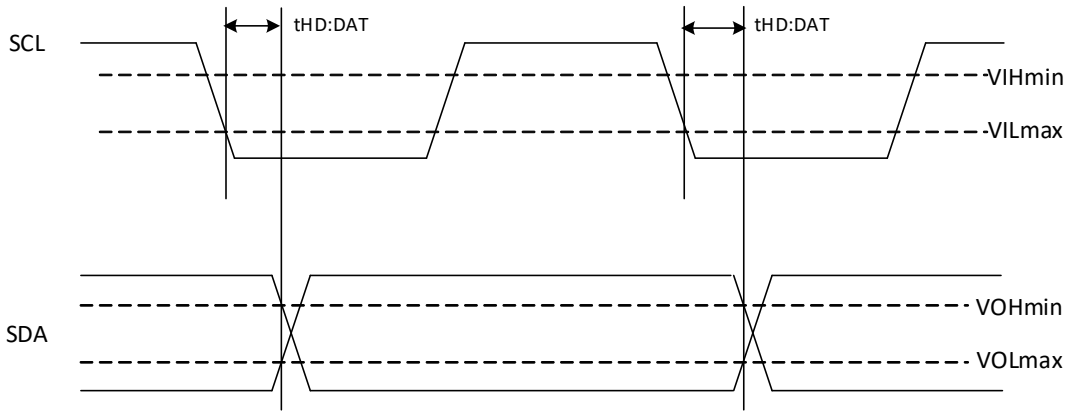


Figure 12 — I²C Bus AC Data Output Timing Parameter Definition

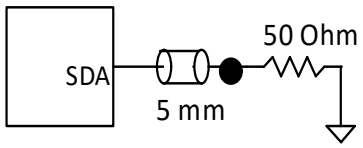


Figure 13 — Output Slew Rate and Output Timing Reference Load

8.1 I²C or I3C Basic Bus Timing (cont'd)

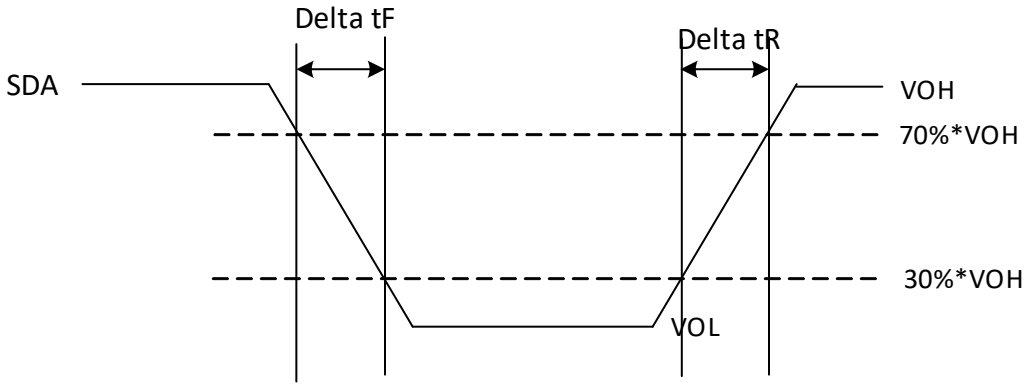


Figure 14 — Output Slew Rate Measurement Points

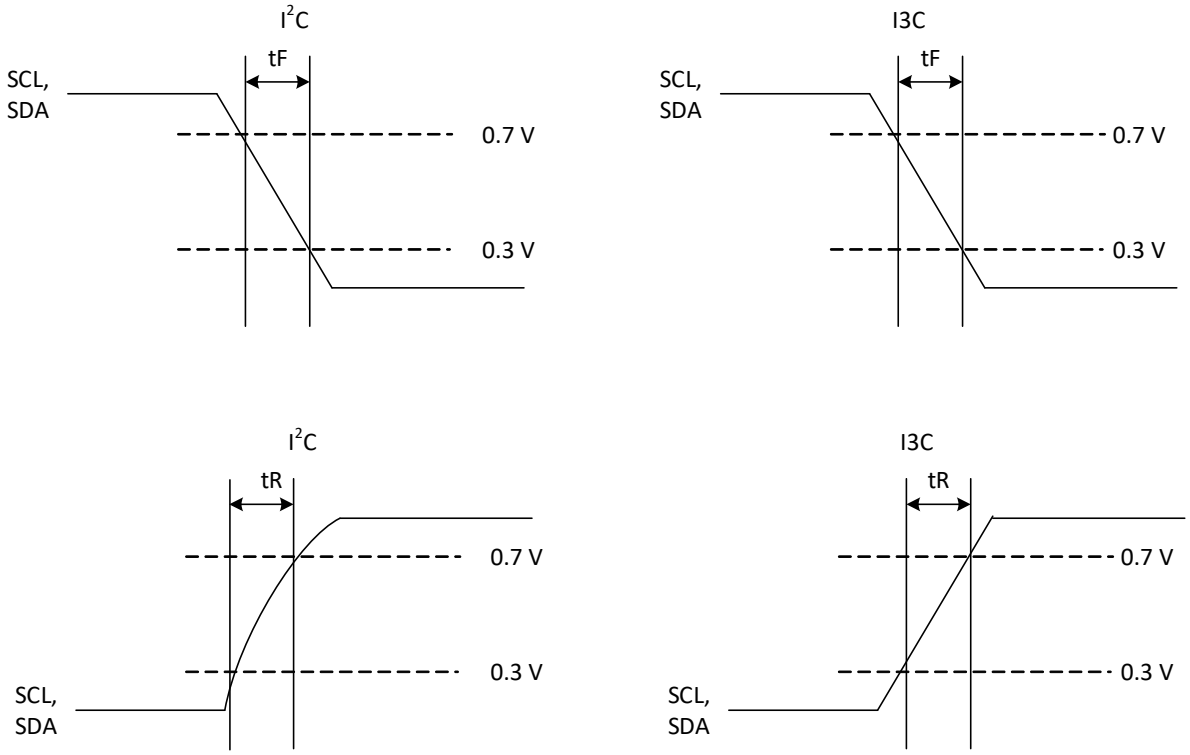


Figure 15 — Rise and Fall Timing Parameter Definition

9 Parametric Characteristics

9.1 Absolute Maximum Ratings

Stressing the device above the rating listed in the Absolute Maximum Ratings table may cause permanent damage to the device. These are stress ratings only and operation of the device at these or any other conditions above those indicated in the Operating sections of this specification is not implied. Exposure to Absolute Maximum Rating conditions for extended periods may affect device reliability

.

Table 58 — Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Units
T_{STG}	Storage temperature	-65	150	°C
V_{DDIO}	Supply voltage	-0.5	2.1	V
V_{DDSPD}	Supply voltage	-0.5	2.1	V
SA	SA Pin	-0.5	2.1	V
SCL, SDA	SCL, SDA Pins (Grade A)	-0.5	3.6	V
SCL, SDA	SCL, SDA Pins (Grade B)	-0.5	2.1	V

9.2 Operating Condition, Measurement Condition, and DC and AC Characteristics

This section summarizes the operating and measurement conditions, and the DC and AC characteristics of the device. The parameters in the DC and AC Characteristic tables that follow are derived from tests performed under the Measurement Conditions summarized in the relevant tables.

Table 59 — DC Operating Conditions

Symbol	Parameter	Min	Typ	Max	Units
V_{DDSPD}	Input Supply Voltage[1]	1.7	1.8	1.98	V
V_{DDIO}	Input Supply Voltage for IO[2]	0.95	1.0	1.05	V
T_{CASE}	Case operating temperature	-40		125	°C

NOTE 1 For DDR5 DIMM application, the DDR5 PMIC VOUT1.8V setting should be selected such that absolute Min and Max values for TS spec are not violated

NOTE 2 For DDR5 DIMM application, the DDR5 PMIC VOUT1.0V setting should be selected such that absolute Min and Max values for TS spec are not violated.

9.2 Operating Condition, Measurement Condition, DC and AC Characteristics (cont'd)

Table 60 — AC Measurement Conditions[1]

Symbol	Parameter	Min	Max	Units
C_L	Load capacitance	40		pF
	Input rise and Fall times - Open Drain	-	TBD	ns
	Input rise and fall times - Push Pull	-	TBD	ns
	Input signal swing levels	0.2 to 0.8		V
	Input and Output timing reference levels	0.3 to 0.7		V
NOTE 1 This AC measurement condition (Table 60 and Figure 16) is only for the test purpose in lab.				

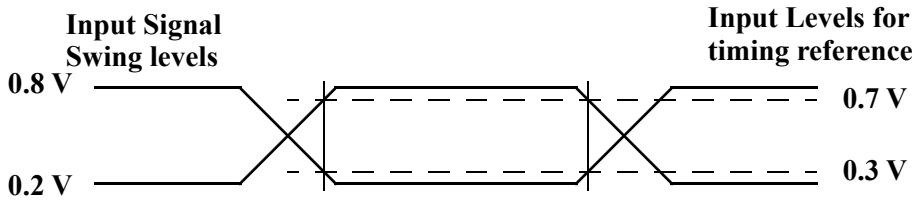


Figure 16 — AC Measurement Waveform

Table 61 — Input Parameters

Symbol	Parameter[1,2]	Test Condition	Min	Max	Units
C_{IN}	Input capacitance (SDA, SCL)	--	--	4	pF
t_{SP}	Pulse width of spikes which must be suppressed by the input filter in I^2C mode.	Single glitch, f ≤ 100 KHz	--	--	ns
		Single glitch, f > 100 KHz	0	50	
NOTE 1 T_A = 25 °C, f = 400 kHz NOTE 2 Verified by design and characterization, not necessarily tested on all devices					

Table 62 — Output Ron Spec

Symbol	Parameter	Min	Max	Units	Notes
R_{on}	SDA, Output Pullup and Pulldown Driver Impedance	20	100	Ohm	1
NOTE 1 Pulldown Ron = Vout/Iout; Pullup Ron = (VDDIO - Vout)/Iout					

9.2 Operating Condition, Measurement Condition, DC and AC Characteristics (cont'd)

Table 63 — DC Characteristics

Symbol	Parameter	Test Condition	Min	Max	Unit
I_{LI}	Input leakage current		--	±5	µA
I_{LO}	Output leakage current		--	±5	µA
I_{DDR}	Supply current, read operation	VDDSPD = 1.8 V, f_C = 12.5 MHz[1]	--	TBD	mA
I_{DDW}	Supply current, write operation	VDDSPD = 1.8V, f_C = 12.5 MHz[1]	--	TBD	mA
I_{DD1}	Standby Supply current	V_{IN} = VDDSPD = 1.8 V	--	TBD	µA
V_{IL}	Input low voltage	--	-0.3	0.3	V
V_{IH}	Input high voltage - Grade A	--	0.7	3.6	V
	Input high voltage - Grade B	--	0.7	1.35	V
V_{OL}	Output low voltage	3 mA sink current	--	0.3	V
V_{OH}	Output high voltage	3 mA source current	0.75	-	V
I_{OL}	Output low current (SDA)	V_{OL} = 0.3 V	3	-	mA
I_{OH}	Output high current (SDA)	V_{OH} = V_{DDIO} - 0.3 V	-	-3	mA
Slew_Rate	Rising Output Slew Rate (SDA)	See NOTE 2	0.1	1.0	V/ns
	Falling Output Slew Rate (SDA)		0.1	3.0	V/ns
V_{PON}	Power On Reset threshold	Monotonic rise between V_{PON} and V_{DDSPD}(min) without ringback	1.6	--	V
V_{POFF}	Power Off threshold for warm power on cycle	No ringback above V_{POFF}	--	0.3	V

NOTE 1 Thermal sensor is active.

NOTE 2 Output slew rate is guaranteed by design and/or characterization. The output slew rate reference load is shown in Figure 13 and Figure 14 shows the timing measurement points. For slew rate measurements, the V_{OH} level shown in Figure 14 is a function of R_{on} value; V_{OH} = {1.0/(R_{on} + 50)} * 50.

9.2 Operating Condition, Measurement Condition, DC and AC Characteristics (cont'd)

Table 64 — AC Characteristics

Symbol	Parameter	I^2C Mode - Open Drain		I3C Basic Push-Pull[1]		Units	Notes
		Min	Max	Min	Max		
f_{SCL}	Clock frequency	0.01	1	0	12.5	MHz	
t_{HIGH}	Clock pulse width high time	260	--	35	--	ns	
t_{LOW}	Clock pulse width low time	500	--	35	--	ns	
$t_{TIMEOUT}$	Detect clock low timeout	10	50	10	50	ms	
t_R	SDA rise time	--	120	--	5	ns	2,3
t_F	SDA fall time	--	120	--	5	ns	2,3
$t_{SU:DAT}$	Data in setup time	50	--	8	--	ns	2
$t_{HD:DI}$	Data in hold time	0		3	--	ns	2
$t_{SU:STA}$	Start condition setup time	260	--	12	--	ns	2
$t_{HD:STA}$	Start condition hold time	260	--	30	--	ns	2
$t_{SU:STO}$	Stop condition setup time	260	--	12	--	ns	2
t_{BUF}	Time between Stop Condition and next Start Condition	500	--	500	--	ns	2,4
t_{POFF}	Warm power cycle off time	1	--	1	--	ms	
t_{Sense_SA}	Time from valid 1.8V supply to Sense SA pin for LID code assignment	-	5	-	5	ms	
t_{INIT}	Time from power on to first command	10	--	10	--	ms	
t_{RST}	Device reinitialization time			TBD	TBD	ms	
t_{AVAL}	Bus Available time (no edges seen on SDA and SCL)	-	-	1	--	µs	
t_{IBI_Issue}	Time to issue IBI after an event is detected when Bus is available	-	-	-	15	µs	
$t_{CLR\_I3C\_CMD\_Delay}$	Time from Clear Register Status to any I3C operation with Start condition to avoid IBI generation; PEC disabled	-	-	4	-	µs	
	Time from Clear Register Status to any I3C operation with Start condition to avoid IBI generation; PEC enabled	-	-	15	-	µs	
$t_{HD:DAT}$	SCL Falling Clock In to SDA Data Out Hold Time	0.5	350	N/A	N/A	ns	5
t_{DOUT}	SCL Falling Clock In to SDA Valid Data Out Time	N/A	N/A	0.5	12	ns	6
t_{DOFFS}	SCL Rising Clock In to SDA Output Off	N/A	N/A	0.5	12	ns	7
t_{DOFFM}	SCL Rising Clock In to Controller SDA Output Off	N/A	N/A	0.5	30	ns	8

Table 64 — AC Characteristics (cont'd)

Symbol	Parameter	I^2C Mode - Open Drain		I3C Basic Push-Pull[1]		Units	Notes
		Min	Max	Min	Max		
$t_{CL\_r\_DAT\_f}$	SCL Rising Clock In to Controller Driving SDA Signal Low	N/A	N/A	40	-	ns	9
$t_{DEVCTRLCCC\_PEC\_DIS}$	DEVCTRL CCC Followed by DEVCTRL CCC or Register Read/Write Command Delay	3	-	3	-	µs	10, 11, 12
$t_{WR\_RD\_DELAY\_PEC\_EN}$	Register Write Command Followed by Register Read Command Delay in PEC Enabled Mode	N/A	N/A	8	-	µs	13, 14, 15
$t_{I2C\_CCC\_Update\_Delay}$	SETHID CCC or SETAASA CCC to any other CCC or Read/Write Command delay	2.5	-	-	-	µs	
$t_{I3C\_CCC\_Update\_Delay}$	RSTDAA CCC or ENEC CCC or DISEC CCC to any other CCC or Read/Write Command Delay	N/A	N/A	2.5	-	µs	
t_{CCC_Delay}	Any CCC to RSTDAA CCC delay	N/A	N/A	2.5	-	µs	

NOTE 1 I3C mode with Open Drain operation follows timing values as shown in I^2C Mode - Open Drain column.

NOTE 2 See Figure 10 for input timing parameter definition.

NOTE 3 See Figure 15 for voltage threshold definition for rise and fall times.

NOTE 4 If PEC is enabled, $t_{WR\_RD\_DELAY\_PEC\_EN}$ timing parameter also applies.

NOTE 5 See Figure 12 for output timing parameter definition

NOTE 6 The TS5 device must be configured in I3C mode to guarantee t_{DOUT} value. See Figure 11 for output timing parameter definition. See Figure 13 for output timing parameter measurement reference load.

NOTE 7 The TS5 device must be configured in I3C mode to guarantee t_{DOFFS} value. See Figure 2. See Figure 13 for output timing parameter measurement reference load.

NOTE 8 The TS5 device must be configured in I3C mode. The Controller guarantees t_{DOFFM} value. See Figure 3. See Figure 13 for output timing parameter measurement reference load.

NOTE 9 See Figure 5.

NOTE 10 From STOP condition of DEVCTRL CCC to START condition for Register Read or Register Write Command Data Packet delay.

NOTE 11 The device sends NACK if Controller does not satisfy $t_{DEVCTRLCCC\_DELAY\_PEC\_DIS}$ timing parameter.

NOTE 12 This timing parameter restriction is only applicable when PEC function is disabled in TS5. If PEC is enabled, this timing parameter does not apply.

NOTE 13 From STOP condition for Register Write Command Data Packet to START condition for Register Read Command Data Packet delay.

NOTE 14 This timing parameter restriction is only applicable when PEC function is enabled in TS5. If PEC is disabled, this timing parameter does not apply.

NOTE 15 The TS5 sends NACK if Controller does not satisfy $t_{WR\_RD\_DELAY\_PEC\_EN}$ timing parameter.

9.3 Temperature Sensor Performance

Table 65 — Temperature Sensor Performance

Parameter	Min	Typ	Max	Unit	Test Conditions
Temperature Sensor Accuracy (Active Range)	--	0.5	1.0	°C	75 °C ≤ T_A ≤ 95 °C
Temperature Sensor Accuracy (Monitor Range)	--	1.0	2.0	°C	40 °C ≤ T_A ≤ 125 °C
Temperature Sensor Accuracy (Industrial Temperature Range)	--	2.0	3.0	°C	-40 °C ≤ T_A ≤ 125 °C
Resolution		0.25		°C	
Conversion Time			125	ms	Assumes 0.25 °C accuracy
Hysteresis between temperature events	1	-	-	°C	

10 Device Package and Pinout

The TS5 device is packaged in 6 ball WLCSP. Ball pitch is 0.5 mm. Package size is 0.8 ± 0.05 mm x 1.3 ± 0.05 mm. The ball diameter is 0.22 mm (after ball attach reflow) and maximum package thickness including the bump is 0.6 mm.

10.1 Package Pinout

The pinout for TS521X is shown in Figure 17. The pinout is a TOP view.

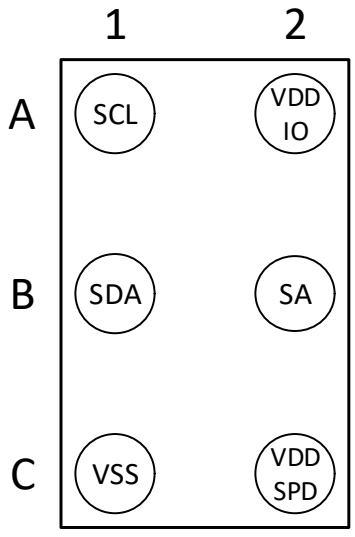


Figure 17 — Pinout for TS5 Device - TOP View

10.2 Package Mechanical Outline (Bottom View)

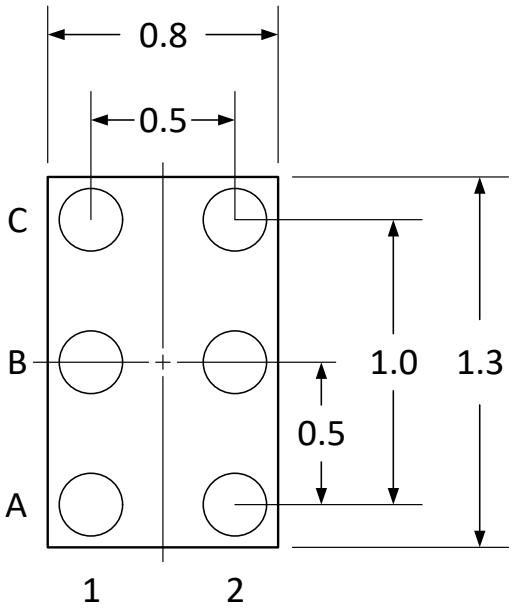
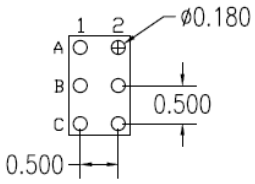


Figure 18 — Package Mechanical Outline

10.3 Recommended PCB Land Pattern



RECOMMENDED LAND PATTERN DIMENSION

NOTES:
1. ALL DIMENSIONS ARE IN MM, ANGLES IN DEGREES.
2. TOP DOWN VIEW, AS VIEW ON PCB.
3. NSMD LAND PATTERN ASSUMED.
4. LAND PATTERN RECOMMENDATION AS PR IPC-7351
   GENERIC REQUIREMENT FOR SURFACE MOUNT DESIGN AND LAND PATTERN.

Figure 19 — Recommended PCB Land Pattern

11 Volatile Registers Space

11.1 Access Mechanism

11.1.1 Register Attribute Definition

All volatile registers have Base Attributes as defined in Table 66. Some register attributes are further modified with Attribute Modifiers, as defined in Table 67.

The volatile register space has a continuous address.

Table 66 — Register Base Attributes

Attribute	Abbreviation	Description
Read Only	R	This bit can be read by software. Writes have no effect.
Read/Write	RW	This bit can be read or written by software.
Write Only	W	This bit can only be written by software.
Reserved	RV	This bit is reserved for future expansion and its value must not be modified by software. The bit will return ‘0’ when read. When writing this bit, software must preserve the value read unless otherwise indicated.

Table 67 — Register Attribute Modifier

Attribute	Abbreviation	Description
Write 1 Only	1O	This bit can only be set (i.e., write ‘1’) but not reset (i.e., write ‘0’)
Protected	P	This bit is protected by the password registers TBD. This bit cannot be written to unless the password code has been written into the password registers.
Persistent	E	Persistent.

11.2 Registers

11.2.1 Register Map

Table 68 — Register Map

Register Name	Register Address (hex)	Attribute	Description
Table 71, “MR0”	0x00	ROE	Device Type; Most Significant Byte
Table 72, “MR1”	0x01	ROE	Device Type; Least Significant Byte
Table 73, “MR2”	0x02	ROE	Device Revision
Table 74, “MR3”	0x03	ROE	Vendor ID Byte 0
Table 75, “MR4”	0x04	ROE	Vendor ID Byte 1
MR5 to MR6	0x05 to 0x06	RV	Reserved
Table 76, “MR7”	0x07	RW	Device Configuration - HID
MR8 to MR17	0x08 to 0x11	RV	Reserved
Table 77, “MR18”	0x12	RW, RO	Device Configuration
Table 78, “MR19”	0x13	1O	Clear Register MR51 Temperature Status Command
Table 79, “MR20”	0x14	1O	Clear Register MR52 Error Status Command
MR21 to MR25	0x15 to 0x19	RV	Reserved
Table 80, “MR26”	0x1A	RW	TS Configuration
Table 81, “MR27”	0x1B	1O, RO, RW	Interrupt Configurations
Table 82, “MR28”	0x1C	RW	TS Temp High Limit Configuration - Low Byte
Table 83, “MR29”	0x1D	RW	TS Temp High Limit Configuration - High Byte
Table 84, “MR30”	0x1E	RW	TS Temp Low Limit Configuration - Low Byte
Table 85, “MR31”	0x1F	RW	TS Temp Low Limit Configuration - High Byte
Table 86, “MR32”	0x20	RW	TS Critical Temp High Limit Configuration - Low Byte
Table 87, “MR33”	0x21	RW	TS Critical Temp High Limit Configuration - High Byte
Table 88, “MR34”	0x22	RW	TS Critical Temp Low Limit Configuration - Low Byte
Table 89, “MR35”	0x23	RW	TS Critical Temp Low Limit Configuration - High Byte
MR36 to MR47	0x24 to 0x2F	RV	Reserved for Device Configuration Type of Registers
Table 90, “MR48”	0x30	RO	Device Status
Table 91, “MR49”	0x31	RO	TS Current Sensed Temperature - Low Byte
Table 92, “MR50”	0x32	RO	TS Current Sensed Temperature - High Byte
Table 93, “MR51”	0x33	RO	TS Temperature Status
Table 94, “MR52”	0x34	RO	Misc. Error Status
MR53 to MR79	0x35 to 0x4F	RV	Reserved
Table 95, “MR80”	0x50	ROE	Serial Number Byte 0[1]
Table 96, “MR81”	0x51	ROE	Serial Number Byte 1[1]
Table 97, “MR82”	0x52	ROE	Serial Number Byte 2[1]
Table 98, “MR83”	0x53	ROE	Serial Number Byte 3[1]
Table 99, “MR84”	0x54	ROE	Serial Number Byte 4[1]
MR85 to MR127	0x55 to 0x7F	RV	Reserved
MR128 to MR255	0x80 to 0xFF	RV	Reserved for Vendor Specific Registers
NOTE 1 Only applicable for the TS521X. This field is reserved in the TS511X.			

11.2.2 Thermal Sensor Registers Read Out Mechanism

All thermal registers are sixteen bit quantities stored in two consecutive registers; low byte first and then high byte. Five bits are reserved for future use. Reserved bits are Read Only bits and must be set to '0' when Controller writes the low byte and high byte. The device always returns '0' in Reserved bits when Controller reads from the low and high byte. Remaining eleven bits in these paired register form a signed value of multiples of 0.25 ranging from -256.00 to + 255.75. Units for all thermal registers are ºC.

The format of each pair of thermal registers is shown in Table 69.

Table 69 — Thermal Register - Low Byte and High Byte

Register		7	6	5	4	3	2	1	0
MRX	Low Byte	8	4	2	1	0.5	0.25	RSVD	RSVD
MRX + 1	High Byte	RSVD	RSVD	RSVD	Sign	128	64	32	16

The examples (reserved bits in grey, sign bit highlighted in cyan) are shown in Table 70.

Table 70 — Thermal Register Examples

High Byte	Low Byte	Value	Unit
000 0 0101	1111 00 00	+95.00	ºC
000 0 0101	0101 00 00	+ 85.00	ºC
000 0 0100	1011 00 00	+ 75.00	ºC
000 0 0000	0001 00 00	+ 1.00	ºC
000 0 0000	0000 11 00	+ 0.75	ºC
000 0 0000	0000 10 00	+ 0.50	ºC
000 0 0000	0000 01 00	+ 0.25	ºC
000 0 0000	0000 00 00	0.00	ºC
000 1 1111	1111 11 00	-0.25	ºC
000 1 1111	1111 10 00	-0.50	ºC
000 1 1111	1111 01 00	-0.75	ºC
000 1 1111	1111 00 00	-1.00	ºC
000 1 1101	1000 00 00	-40.00	ºC

11.2.3 Register Description

Table 71 — MR0

Addr	MR0		Device Type; Most Significant Byte[1]
Bits	Attr	Default	Description
7:0	ROE	-	MR0[7:0]: MSB_DEV_TYPE 0x51: Temp Sensor without Serial Number (TS511X) 0x52: Temp Sensor with Serial Number (TS521X)
NOTE 1 The code in this register is used in conjunction with any device type in Table 72, "MR1" register.			

11.2.3 Register Description (cont'd)

Table 72 — MR1

Addr	MR1		Device Type; Least Significant Byte[1]
Bits	Attr	Default	Description
7:0	ROE	-	MR1[7:0]: LSB_DEV_TYPE Device Type - Temp Sensor - These are hard coded. 0x11 - Grade A Temp Sensor 0x10 - Grade B Temp Sensor
NOTE 1	The code in this register is used in conjunction with any device type in Table 71, "MR0" register.		

Table 73 — MR2

Addr	MR2		Device Revision
Bits	Attr	Default	Description
7:6	RV	0	MR2[7:6]: Reserved
5:4	ROE	-	MR2[5:4]: DEV_REV_MAJOR Major Revision 00 = Revision 1 01 = Revision 2 10 = Revision 3 11 = Revision 4
3:1	ROE	-	MR2[3:1]: DEV_REV_MINOR Minor Revision 000 = Revision 0 001 = Revision 1 010 = Revision 2 .. 111 = Revision 8
0	RV	0	MR2[0]: Reserved

Table 74 — MR3

Addr	MR3		Device Revision
Bits	Attr	Default	Description
7:0	ROE	-	MR3[7:0]: VENDOR_ID_BYTE0 Vendor ID Byte 0

11.2.3 Register Description (cont'd)

Table 75 — MR4

Addr	MR4		Device Revision
Bits	Attr	Default	Description
7:0	ROE	-	MR4[7:0]: VENDOR_ID_BYTE1 Vendor ID Byte 1

Table 76 — MR7

Addr	MR7		Device Configuration[1]
Bits	Attr	Default	Description
7:4	RV	0	MR7[7:4]:Reserved
3:1	RW	111	MR7[7:0]:DEV_HID_CODE Device HID Code. The TS5 device responds to unique 7-bit address as formed by 4 bit LID code as in Table 2 and 3 bit HID code as configured in this register. This register is updated when SETHID CCC is registered by the TS5 device or when TS5 device goes through bus reset as described in Section 6.15.
0	RV	0	MR7[0]: Reserved

NOTE 1 The write (or update) transaction to this register must be followed by STOP operation to allow the TS5 device to update the setting.

11.2.3 Register Description (cont'd)

Table 77 — MR18

Addr	MR18		Device Configuration[1]
Bits	Attr	Default	Description
7	RW	0	MR18[7]: PEC_EN PEC Enable[2,3] 0 = Disable 1 = Enable
6	RW	0	MR18[6]: PAR_DIS Parity (T bit) Disable[3,4] 0 = Enable 1 = Disable
5	RO	0	MR18[5]: INF_SEL Interface Selection 0 = I^2C Protocol (Max speed of 1 MHz) 1 = I3C Basic Protocol[5]
4	RW	0	MR18[4]: DEF_RD_ADDR_POINT_EN Default Read Address Pointer Enable 0 = Disable Default Read Address Pointer (Address pointer is set by the Controller)[6] 1 = Enable Default Read Address Pointer; Address selected by register bits [3:2]
3:2	RW	0	MR18[3:2]: DEF_RD_ADDR_POINT_START Default Read Pointer Starting Address[7] 00 = Table 91, "MR49" 01 = Reserved 10 = Reserved 11 = Reserved
1	RW	0	MR18[1]: DEF_RD_ADDR_POINT_BL Burst Length for Read Pointer Address for PEC Calculation[8] 0 = 2 Bytes 1 = 4 Bytes
0	RV	0	MR18[0]: Reserved

NOTE 1 The write (or update) transaction to this register must be followed by STOP operation to allow the TS5 device to update the setting.

NOTE 2 This register is only applicable if Table 77, "MR18" [5] = '1'.

NOTE 3 This register is updated when RSTDAA CCC is registered by the TS5 device or when or when TS5 device goes through bus reset as described in Section 6.15.

NOTE 4 This register is only applicable if Table 77, "MR18" [5] = '1'. When Parity function is disabled, the TS5 device simply ignores the "T" bit information from the Controller. The controller may actually choose to compute the parity and send that information in "T" bit or simply drive static low or high in "T" bit.

NOTE 5 This register is automatically updated when SETAASA CCC or RSTDAA CCC is registered by the TS5 device or when TS5 device goes through bus reset as described in Section 6.15. This register can be read by the Controller through normal read operation but cannot be written with normal write operation either in I^2C mode or I3C Basic mode. When this register is updated, it takes in effect when there is a next START operation (i.e., after STOP operation).

NOTE 6 The setting in registerTable 77, "MR18" [3:1] is a don't care.

NOTE 7 This register is only applicable if Table 77, "MR18" [4] = '1'.

NOTE 8 This register is only applicable if Table 77, "MR18" [7, 4] = '11'.

11.2.3 Register Description (cont'd)

Table 78 — MR19

Addr	MR19		Clear Register Command[1]
Bits	Attr	Default	Description
7:4	RV	0	MR19[7:4]: Reserved
3	1O	0	MR19 [3]: CLR_TS_CRIT_LOW Clear Temp Sensor Critical Low Status 1 = Clear Table 93, "MR51" [3] Register
2	1O	0	MR19 [2]: CLR_TS_CRIT_HIGH Clear Temp Sensor Critical High Status 1 = Clear Table 93, "MR51" [2] Register
1	1O	0	MR19 [1]: CLR_TS_LOW Clear Temp Sensor Low Status 1 = Clear Table 93, "MR51" [1] Register
0	1O	0	MR19 [0]: CLR_TS_HIGH Clear Temp Sensor High Status 1 = Clear Table 93, "MR51" [0] Register
NOTE 1 This entire register is self clearing register after corresponding register is cleared.			

Table 79 — MR20

Addr	MR20		Clear Register Command[1]
Bits	Attr	Default	Description
7:2	RV	0	MR20[7:2]: Reserved
1	1O	0	MR20[1]: CLR_PEC_ERROR Clear Packet Error Status 1 = Clear Table 94, "MR52" [1] Register
0	1O	0	MR20[0]: CLR_PAR_ERROR Clear Parity Error Status 1 = Clear Table 94, "MR52" [0] Register
NOTE 1 This entire register is self clearing register after corresponding register is cleared.			

Table 80 — MR26

Addr	MR26		Thermal Sensor Configuration
Bits	Attr	Default	Description
7:1	RV	0	MR26[7:1] Reserved
0	RW	0	MR26[0]: DIS_TS Disable Temp Sensor[1] 0 = Enable thermal sensor 1 = Disable thermal sensor
NOTE 1 If this bit is set to '1' and then reset to '0', the controller must wait minimum of tINIT before accessing samples on the thermal sensor.			

11.2.3 Register Description (cont'd)

Table 81 — MR27

Addr	MR27		Interrupt Configuration
Bits	Attr	Default	Description
7	1O	0	MR27[7]: CLR_GLOBAL Global Clear Event Status and In Band Interrupt Status[1,2] 1 = Clear Table 90, "MR48" [7], Table 93, "MR51" [3:0] and Table 94, "MR52" [1:0] Register
6:5	RV	0	MR27[6:5]: Reserved
4	RO	0	MR27 [4]: IBI_ERROR_EN In Band Error Interrupt Enable for Table 94, "MR52" Error Log[3] 0 = Disable; Errors logged in Table 94, "MR52" [1:0] registers do not generate an IBI to Controller 1 = Enable; Errors logged in Table 94, "MR52" [1:0] registers generates an IBI to Controller
3	RW	0	MR27 [3]: IBI_TS_CRIT_LOW_EN In Band Error Interrupt Enable for Temp Sensor Critical Low 0 = Disable; Table 93, "MR51"[3] = '1' does not generate an IBI to Controller 1 = Enable; Table 93, "MR51"[3] = '1' and Table 81, "MR27" [4] = '1' generates an IBI to Controller
2	RW	0	MR27[2]: IBI_TS_CRIT_HIGH_EN In Band Error Interrupt Enable for Temp Sensor Critical High 0 = Disable; Table 93, "MR51"[2] = '1' does not generate an IBI to Controller 1 = Enable; Table 93, "MR51"[2] = '1' and Table 81, "MR27" [4] = '1' generates an IBI to Controller
1	RW	0	MR27[1]: IBI_TS_LOW_EN In Band Error Interrupt Enable for Temp Sensor Low 0 = Disable; Table 93, "MR51"[1] = '1' does not generate an IBI to Controller 1 = Enable; Table 93, "MR51"[1] = '1' and Table 81, "MR27" [4] = '1' generates an IBI to Controller
0	RW	0	MR27[0]: IBI_TS_HIGH_EN In Band Error Interrupt Enable for Temp Sensor High 0 = Disable Table 93, "MR51"[0] = '1' does not generate an IBI to Controller 1 = Enable; Table 93, "MR51"[0] = '1' and Table 81, "MR27" [4] = '1' generates an IBI to Controller

NOTE 1 This register is a self clearing register after corresponding registers are cleared. Writing '0' in this register has no effect.

NOTE 2 After this command is issued, the device does not generate an IBI for any pending event. But if new event occurs, the device does generate an IBI.

NOTE 3 This register is automatically updated when ENEC CCC or DISEC CCC or RSTDAA CCC is registered by the TS5 device or when TS5 device goes through bus reset as described in Section 6.15. This register can be read by the Controller through normal read operation but cannot be written with normal write operation either in I^2C mode or I3C Basic mode. When this register is updated, it takes effect when there is a next START operation (i.e., after STOP operation).

11.2.3 Register Description (cont'd)

Table 82 — MR28

Addr	MR28		Thermal Sensor High Limit Configuration - Low Byte[1,2]
Bits	Attr	Default	Description
7:0	RW	0x70	MR28[7:0]: TS_HIGH_LIMIT_LOW Table 82, "MR28" and Table 83, "MR29" - 16 bit thermal registers define the high limit for thermal sensor. See Table 69, "Thermal Register - Low Byte and High Byte".
NOTE 1	Critical temperature High Limit value must have a higher value than temperature High Limit (Table 82, "MR28" [7:0] and Table 83, "MR29" [7:0].		
NOTE 2	The Reserved bits are Read Only bits. The controller must write '0' in reserved bit when writing and device always returns '0' from reserved bits when controller reads this byte.		

Table 83 — MR29

Addr	MR29		Thermal Sensor High Limit Configuration - High Byte[1,2]
Bits	Attr	Default	Description
7:0	RW	0x03	MR29[7:0]: TS_HIGH_LIMIT_HIGH Table 82, "MR28" and Table 83, "MR29" - 16 bit thermal registers define the high limit for thermal sensor. See Table 69, "Thermal Register - Low Byte and High Byte".
NOTE 1	Critical temperature High Limit value must have a higher value than temperature High Limit (Table 82, "MR28" [7:0] and Table 83, "MR29" [7:0].		
NOTE 2	The Reserved bits are Read Only bits. The controller must write '0' in reserved bit when writing and device always returns '0' from reserved bits when controller reads this byte.		

Table 84 — MR30

Addr	MR30		Thermal Sensor Low Limit Configuration - Low Byte[1,2]
Bits	Attr	Default	Description
7:0	RW	0	MR30[7:0]: TS_LOW_LIMIT_LOW Table 84, "MR30" and Table 85, "MR31" - 16 bit thermal registers define the low limit for thermal sensor. See Table 69, "Thermal Register - Low Byte and High Byte"
NOTE 1	Critical temperature Low Limit value must have a lower value than temperature Low Limit (Table 84, "MR30" [7:0] and Table 85, "MR31" [7:0]		
NOTE 2	The Reserved bits are Read Only bits. The controller must write '0' in reserved bit when writing and device always returns '0' from reserved bits when controller reads this byte.		

Table 85 — MR31

Addr	MR31		Thermal Sensor Low Limit Configuration - High Byte[1,2]
Bits	Attr	Default	Description
7:0	RW	0	MR31[7:0]: TS_LOW_LIMIT_HIGH Table 84, "MR30" and Table 85, "MR31" - 16 bit thermal registers define the low limit for thermal sensor. See Table 69, "Thermal Register - Low Byte and High Byte"
NOTE 1	Critical temperature Low Limit value must have a lower value than temperature Low Limit (Table 84, "MR30" [7:0] and Table 85, "MR31" [7:0].		
NOTE 2	The Reserved bits are Read Only bits. The controller must write '0' in reserved bit when writing and device always returns '0' from reserved bits when controller reads this byte.		

11.2.3 Register Description (cont'd)

Table 86 — MR32

Addr	MR32		Thermal Sensor Critical Temperature High Limit Configuration - Low Byte[1,2]
Bits	Attr	Default	Description
7:0	RW	0x50	MR32[7:0]: TS_CRIT_HIGH_LIMIT_LOW Table 86, "MR32" and Table 87, "MR33" - 16 bit thermal registers define the critical temperature high limit for thermal sensor. See Table 69, "Thermal Register - Low Byte and High Byte"
NOTE 1	Critical temperature High Limit value must have a higher value than temperature High Limit (Table 82, "MR28" [7:0] and Table 83, "MR29" [7:0].		
NOTE 2	The Reserved bits are Read Only bits. The controller must write '0' in reserved bit when writing and device always returns '0' from reserved bits when controller reads this byte.		

Table 87 — MR33

Addr	MR33		Thermal Sensor Critical Temperature High Limit Configuration- High Byte[1,2]
Bits	Attr	Default	Description
7:0	RW	0x05	MR33[7:0]: TS_CRIT_HIGH_LIMIT_HIGH Table 86, "MR32" and Table 87, "MR33" - 16 bit thermal registers define the critical temperature high limit for thermal sensor. See Table 69, "Thermal Register - Low Byte and High Byte"
NOTE 1	Critical temperature High Limit value must have a higher value than temperature High Limit (Table 82, "MR28" [7:0] and Table 83, "MR29" [7:0].		
NOTE 2	The Reserved bits are Read Only bits. The controller must write '0' in reserved bit when writing and device always returns '0' from reserved bits when controller reads this byte.		

Table 88 — MR34

Addr	MR34		Thermal Sensor Critical Temperature Low Limit Configuration- Low Byte[1,2]
Bits	Attr	Default	Description
7:0	RW	0	MR34[7:0]: TS_CRIT_LOW_LIMIT_LOW Table 88, "MR34" and Table 89, "MR35" - 16 bit thermal registers define the critical temperature low limit for thermal sensor. See Table 69, "Thermal Register - Low Byte and High Byte"
NOTE 1	Critical temperature Low Limit value must have a lower value than temperature Low Limit (Table 84, "MR30" [7:0] and Table 85, "MR31" [7:0].		
NOTE 2	The Reserved bits are Read Only bits. The controller must write '0' in reserved bit when writing and device always returns '0' from reserved bits when controller reads this byte.		

11.2.3 Register Description (cont'd)

Table 89 — MR35

Addr	MR35		Thermal Sensor Critical Temperature Low Limit Configuration- High Byte[1,2]
Bits	Attr	Default	Description
7:0	RW	0	MR35[7:0]: TS_CRIT_LOW_LIMIT_HIGH Table 88, "MR34" and Table 89, "MR35" - 16 bit thermal registers define the critical temperature low limit for thermal sensor. See Table 69, "Thermal Register - Low Byte and High Byte"
NOTE 1	Critical temperature High Limit value must have a higher value than temperature High Limit (Table 84, "MR30" [7:0] and Table 85, "MR31" [7:0].		
NOTE 2	The Reserved bits are Read Only bits. The controller must write '0' in reserved bit when writing and device always returns '0' from reserved bits from reserved bits when controller reads this byte		

Table 90 — MR48

Addr	MR48		Device Status
Bits	Attr	Default	Description
7	RO	0	MR48[7]: IBI_STATUS Device Event In Band Interrupt Status 0 = No pending IBI 1 = Pending IBI
6:0	RV	0	MR48[6:0]: Reserved

Table 91 — MR49

Addr	MR49		Current Sensed Temperature - Low Byte[1]
Bits	**Attr**	**Default**	**Description**
7:0	RO	0	MR49[7:0]: TS_SENSE_LOW Table 91, "MR49" and Table 92, "MR50" - 16 bit thermal registers return the most recent conversion of the thermal sensor See Table 69, "Thermal Register - Low Byte and High Byte"
NOTE 1	The device always returns '0' from reserved bits when controller reads this byte.		

Table 92 — MR50

Addr	MR50		Current Sensed Temperature - High Byte[1]
Bits	Attr	Default	Description
7:0	RO	0	MR50[7:0]: TS_SENSE_HIGH Table 91, "MR49" and Table 91, "MR49" - 16 bit thermal registers return the most recent conversion of the thermal sensor See Table 69, "Thermal Register - Low Byte and High Byte"
NOTE 1	The device always returns '0' from reserved bits when controller reads this byte.		

11.2.3 Register Description (cont'd)

Table 93 — MR51

Addr	MR51		Thermal Sensor Temperature Status
Bits	Attr	Default	Description
7:4	RV	0	MR51[7:4]: Reserved
3	RO	0	MR51[3]: TS_CRIT_LOW_STATUS Temp Sensor Critical Low 0 = Temperature is above the limit set in Table 88, "MR34" and Table 89, "MR35" 1 = Temperature is below the limit set in Table 88, "MR34" and Table 89, "MR35"
2	RO	0	MR51[2]: TS_CRIT_HIGH_STATUS Temp Sensor Critical High 0 = Temperature is below the limit set in Table 86, "MR32" and Table 87, "MR33" 1 = Temperature is above the limit set in Table 86, "MR32" and Table 87, "MR33"
1	RO	0	MR51[1]: TS_LOW_STATUS Temp Sensor Low 0 = Temperature above limit set in registers Table 84, "MR30" and Table 85, "MR31" 1 = Temperature below limit set in registers Table 84, "MR30" and Table 85, "MR31"
0	RO	0	MR51[0]: TS_HIGH_STATUS Temp Sensor High 0 = Temperature is below the limit set in registers Table 82, "MR28" and Table 83, "MR29" 1 = Temperature is above the limit set in registers Table 82, "MR28" and Table 83, "MR29"

Table 94 — MR52

Addr	MR52		Misc. Error Status
Bits	Attr	Default	Description
7:2	RV	0	MR52[7:2]: Reserved
1	RO	0	MR52[1]: PEC_ERROR_STATUS Packet Error[1,2] 0 = No PEC Error 1 = PEC Error in one or more packets
0	RO	0	MR52[0]: PAR_ERROR_STATUS Parity Check Error[2,3] 0 = No Parity Error 1 = Parity Error in one or more bytes

NOTE 1 Only applicable Table 77, "MR18" [5] = '1' and if PEC function is enabled.

NOTE 2 This register is updated when TS5 device goes through bus reset as described in Section 6.15.

NOTE 3 Only applicable in Table 77, "MR18" [5] = '1' and if Parity function is not disabled or for supported CCC in I^2C mode.

11.2.3 Register Description (cont'd)

Table 95 — MR80

Addr	MR80		Serial Number Byte 0
Bits	Attr	Default	Description
7:0	ROE	-	MR80[7:0]: SERIAL_NUMBER_BYTE_0[1] Byte 0 of the unique 40-bit serial number stored in Table 95, "MR80" to Table 99, "MR84". The serial number is vendor specific
NOTE 1 Only applicable for the TS521X. This field is reserved in the TS511X.			

Table 96 — MR81

Addr	MR81		Serial Number Byte 1
Bits	Attr	Default	Description
7:0	ROE	-	MR81[7:0]: SERIAL_NUMBER_BYTE_1[1] Byte 1 of the unique 40-bit serial number stored in Table 95, "MR80" to Table 99, "MR84". The serial number is vendor specific
NOTE 1 Only applicable for the TS521X. This field is reserved in the TS511X.			

Table 97 — MR82

Addr	MR82		Serial Number Byte 2
Bits	Attr	Default	Description
7:0	ROE	-	MR82[7:0]: SERIAL_NUMBER_BYTE_2[1] Byte 2 of the unique 40-bit serial number stored in Table 95, "MR80" to Table 99, "MR84". The serial number is vendor specific
NOTE 1 Only applicable for the TS521X. This field is reserved in the TS511X.			

Table 98 — MR83

Addr	MR83		Serial Number Byte 3
Bits	Attr	Default	Description
7:0	ROE	-	MR83[7:0]: SERIAL_NUMBER_BYTE_3[1] Byte 3 of the unique 40-bit serial number stored in Table 95, "MR80" to Table 99, "MR84". The serial number is vendor specific
NOTE 1 Only applicable for the TS521X. This field is reserved in the TS511X.			

11.2.3 Register Description (cont'd)

Table 99 — MR84

Addr	MR84		Serial Number Byte 3
Bits	Attr	Default	Description
7:0	ROE	-	MR84[7:0]: SERIAL_NUMBER_BYTE_4[1] Byte 4 of the unique 40-bit serial number stored in Table 95, "MR80" to Table 99, "MR84". The serial number is vendor specific
NOTE 1 Only applicable for the TS521X. This field is reserved in the TS511X.			

Annex A — (Informative) Differences between JESD302-1.01 and JESD302-1

This annex briefly describes most of the changes made to entries that appear in this standard, JESD302-1.01, compared to its predecessor, JESD302-1, January 2022. This is an editorial change.

Clause	Description of change
1.9.2	Table 63, fixed typo in IOL/IOH values.
11.2.2	Table 70, for -40.00 changed High Byte 000 1 111 to 000 1 1101, and for Low Byte changed 1101 10 00 to 1000 00 00.

Annex B — (Informative) Differences between JESD302-1A and JESD302-1.01

This annex briefly describes most of the changes made to entries that appear in this standard, JESD302-1A, compared to its predecessor, JESD302-1.01, April 2022.

Editorial changes:

1. Terminology update to replace every instance of "master" with "controller" and "slave" with "target"
2. Updated the JEDEC "notice" and "do not violate the law" pages
3. Updated all tables to the JEDEC format; i.e., NOTES are inside cells at the bottom of the table
4. Updated all figure title to appear at the bottom of each figure
5. Added Table of Contents, List of Figures, and List of Tables

Technical changes:

1. Changed document title from "TS5111, TS5110 Serial Bus Thermal Sensor Device Standard" to "TS511X, TS521X Serial Bus Thermal Sensor Device Standard"
2. Restored Figure 5
3. Modified Table 68 to add Register Names for Tables 95 through 99
4. Added Tables 95 through 99
5. Added clarifying notes to serial number registers
6. Added clarifying text to Table 71

Standard Improvement Form **JEDEC Standard JESD302-1A, Version 2.00**

The purpose of this form is to provide the Technical Committees of JEDEC with input from the industry regarding usage of the subject standard. Individuals or companies are invited to submit comments to JEDEC. All comments will be collected and dispersed to the appropriate committee(s).

If you can provide input, please complete this form and return to:

JEDEC
Attn: Publications Department
3103 North 10th Street
Suite 240 South
Arlington, VA 22201-2107

Fax: 703.907.7583

1. I recommend changes to the following:

☐ Requirement, clause number _____

☐ Test method number _____ Clause number _____

The referenced clause number has proven to be:

☐ Unclear ☐ Too Rigid ☐ In Error

☐ Other _____

2. Recommendations for correction:

3. Other suggestions for document improvement:

Submitted by

Name: _____ Phone: _____

Company: _____ E-mail: _____

Address: _____

City/State/Zip: _____ Date: _____


JEDEC®